The ISSA Journal

January 2009
Volume 7 Issue 1

Mastering Data Retention

Cryptography as a Service

Key Information Security Trends for 2009

Complying with the Red Flag Rules

Ways to Determine or Prioritize Security Initiatives

The Dark Side of Computing: The study of computer crime, Part 1

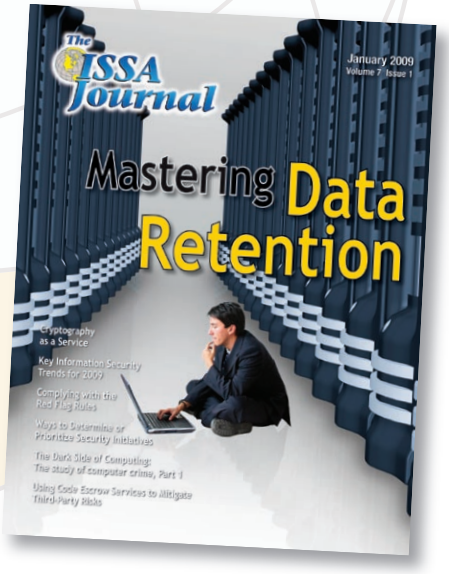
Using Code Escrow Services to Mitigate Third-Party Risks

Table of Contents

Feature

Articles

Also in this issue



The ISSA Journal (USPS PP 152) is published monthly by the Information Systems Security Association, 9220 SW Barbur Blvd. #119-333, Portland, Oregon 97219. Application to mail at periodicals postage rates is pending at Portland, Oregon and at additional mailing offices. **Postmaster**: Send address changes to ISSA Journal, 9220 SW Barbur Blvd., #119-333, Portland, Oregon 97219.

Using Code Escrow Services to Mitigate Third-Party Risks

By Raoul Gomes and Rafael Etges

This article discuses source code escrow and how the service can be used to mitigate the risks associated with SaaS or custom software developed by third parties.

The use of Software-as-a-Service (SaaS) has increased for many reasons: flexibility and productivity gains of telecommuting employees, the reduced cost of ownership for software licenses and maintenance cycles, or the strategic driver of being a lean organization. In addition to the risks associated with any software (misuse of code and others), the risks associated with SaaS or custom software developed by third parties include the lack of access to the source code in the event of a business disruption or security investigation. Source code escrow is an area that is just gaining ground to address these concerns and to mitigate the risks surrounding these scenarios. This article will discuss the code escrow approach and benefits provided by this service.

Escrow service has been around in many industries in order to mitigate risk. It is used to facilitate the transfer of property from one individual to another through the use of an independent third party. Essentially it consists of an agreement that an item is deposited with an escrow agent, held in trust or security, and delivered to the grantee or promisee on the fulfillment of certain future conditions. Individuals are probably most familiar with this service during a real estate transaction, e.g., ensuring the transfer of title once certain conditions have been met such as the passing of inspection.

What is escrow and code escrow?

Most information security practitioners are familiar with encryption key escrow, whereby the custody of a decryption key is held by two or more parties; in order to recover the decryption key, input from these parties is required. Key escrow gained media attention in the U.S. during the Clipper Chip[1] controversy, when in 1993 the Clinton administration proposed a new standard developed with the National Security Agency. Under the standard, computer chips would use an algorithm called Skipjack to encrypt information; the FBI and the Justice Department would have control over the decryption keys. The Clipper Chip device was designed to be installed on all telephones, computer modems, and fax machines to encrypt voice communications. The key recovery process used during criminal investigations would be dependant upon a warrant to prevent inappropriate interception of communications by law enforcement; however, public outcry and privacy concerns prevented the Clipper Chip initiative from being executed in a large scale at the time.

Despite its eventual failure, the Clipper Chip initiative is a definite example of escrow mechanisms being coordinated and deployed on a very large scale. When the specific conditions on this escrow model were met, the decryption keys would be released to the selected entities and communication records would become accessible for law enforcement purposes.

Escrow companies are also commonly used in the transfer of high value personal and business property, like websites and businesses, and in the completion of person-to-person remote auctions (such as eBay). A similar approach can be taken to govern and control the ownership and possession of application source code. Source code escrow agents hold source code of software in trust just as other escrow companies hold cash. The highly valuable (and often secret) source code is only released by the agent to either party upon specific terms of the escrow agreement (such as failure to maintain the application, transfer of ownership of the intellectual property rights, or the liquidation of the owner of the source code).[2]

Why use code escrow?

In recent years the protection of applications and systems has gradually replaced the focus on network security following a maturation of technologies designed to safeguard networks. Once networks became more resilient, attack vectors moved to the application security space and started exploiting vul-

1 More information about the Clipper Chip controversy can be found in the Wikipedia article: http://en.wikipedia.org/wiki/Clipper_Chip. The National Institute of Standards and Technology provides additional documentation and discussion papers about encryption key escrow: http://csrc.nist.gov/keyrecovery.

2 Wikipedia – http://en.wikipedia.org/wiki/Escrow.

nerable code. This evolution to an arms race between attackers and industry demands attention to the availability and integrity of source code.

Outsourcing application development peril

Economic drivers are encouraging the outsourcing of software development in which a significant portion of the code being created by third parties is exposed to risk. Quocirca, a business and IT research group, conducted a study with IT directors and executives from 250 companies across Germany, the UK, and the U.S. who were accountable for the security of corporate applications. All those who admitted to being subjected to frequent hacking had outsourced some level of software development, with almost 90% outsourcing more than 40% of the development.[3]

In Canada, another study performed in conjunction with the University of Toronto in 2008 found that application security breaches are reported more often by organizations using outsourcers than those who do not outsource.[4] For example, 6% of outsourcers reported Web defacements as compared to only 1% for non-outsourcers. Outsourcers also fared slightly worse in the areas of identity theft and misuse of a public Web application. Likely related to lack of application-level controls, breaches relating to loss of confidential data were much higher for outsourcing organizations at 9%, compared to 4% for those that do not outsource.

These studies are showing that organizations relying on third-party software for business critical operations are putting themselves in a significant degree of risk. This can be either through a specific application or SaaS which is used as part of a critical process. Even corporations with the best internal controls in place and well-managed processes will still be exposed as this point of failure is outside of their control, sometimes regulated by contracts and service agreements only, which are not preventative measures and will do little good in the case of a business interruption.

A couple outsourcing nightmares:

- During the summer of 2007 the online trading company TS Ameritrade was forced to disclose a breach involving the personal details regarding 6.3 million customers caused by a back door created by a programmer.[5]
- Another example of how the unlikeliest of circumstances can catch up to any organization relates to IT Factory, a major SaaS provider. On December 1, 2008 the company adjudicated bankruptcy. This was a reputable company that was awarded Denmark's *Best IT-company 2008*.[6] It remains to be seen what the outcome shall be for its 1.3 million users, but one cannot refute the benefits that code escrow would bring to this situation.

Software service providers manage risk the same way enterprises do: they assess the risks to themselves and apply controls accordingly. When the source code of applications is shared with an independent escrow agent and available to the enterprise for ownership transfer, investigation, or litigation under certain conditions (e.g., breach, violation of SLAs, or business interruption of the provider), the provider will be encouraged to apply sufficient safeguards internally to protect itself and its customers. There is now a transfer or sharing, to some extent, of the risks to the service provider that does not happen otherwise. Without an escrow mechanism in place, the enterprise is solely exposed to the risks.

Risk mitigated through code escrow

Consider, from a risk perspective, what would happen in the event of a business interruption by that third party supplier to your operations? What if they go out of business without warning? What if this were a billing software, an inventory management system, or a client management system and it stops functioning? Most business interruption insurance policies do not cover these scenarios, and if they do, the costs can be substantial. The bottom line is that it does not solve the problem but only provides some form of compensation.

In such cases, the escrow mechanism can be your best option to gain access to the source code of these critical applications. Ownership can be restored to the purchasing organization ensuring the continuity of their life cycle. Internal resources or another third party could be employed to maintain these applications and, although there would be a cost associated with the change, the organization would have a choice other than stopping its operations until an alternative system is selected, deployed, and business support is properly transitioned.

How does it work?

Once certain software have been identified as requiring risk mitigation actions, the option of code escrow is available to the organization. The following outlines the process and what to expect when exercising this option.

1. Selection of escrow agent, agreement with software provider, and contract negotiation

During this step a trusted and reliable escrow agent should be selected by the enterprise, and the service provider must agree with the escrow conditions. Details such as regular updates to the source code repository in escrow should be arranged between the service provider and the escrow agent, with some degree of monitoring from the enterprise. Contractual language should be revised carefully with provisions for the conditions under which the source code will be transferred to the enterprise, the service provider maintenance and change management processes, as well as how the escrow agent will

3 A summary of the survey can be found at http://www.quocirca.com/pages/analysis/reports/view/store250/item21107/?link_683=21107.

4 The full TELUS/Rotman School of Business report from the University of Toronto can be found at www.telus.com/securityreport.

5 Quorcica survey as in #2.

6 http://www.computerworld.dk/art/47637/top-100-her-er-danmarks-dygtigste-it-virksomhed?a=fp_3&i=1.

safeguard the source code (e.g., access controls, encryption, physical security, etc).

The enterprise may be involved in ensuring that the source code submitted to the escrow agent is valid: years ago, Radisson Hotels Worldwide outsourced the maintenance of its mission-critical reservation system with an escrow agreement in place. The code was released as a result of the provider going out of business; however, the source code in escrow was missing many components and the escrow account did not contain any documentation developed after the initial escrow of the software.

Advocates *against* code escrow refer to this case as an example of a failure;[7] however, that instance serves to reinforce the fact that the enterprise must take ownership and monitor the development (external or internal) of its critical code with diligence.

2. Deployment of escrow agreement

During regular operations, the enterprise and the escrow agent must be involved in the service provider's change management cycles: the enterprise must be informed of a change (both regular and emergency changes), and the escrow agent must receive the new code to update its libraries.

3. Trigger and execution of the escrow mechanism

If one of the clauses for code transfer is triggered by an event stipulated in the escrow contract, this will probably be detected and communicated by the enterprise to the service provider and the escrow agent. More than likely the service provider will require validation or a chance to further investigate, as it is not in their best interest that the source code is shared or transferred to the enterprise. Depending on the relationship between enterprise and service provider, it may be advisable to involve legal counsel if the matter needs to be expedited. However, since the code is now held by an independent escrow agent, the enterprise may exercise its right to gain access to the code once it has been determined that the event has happened beyond reasonable doubt.

4. The day after

The organization finally gets the source code back. This could be (1) a demand from an incident response procedure, for investigative purposes following a security breach involving the application in escrow, in which case the incident response team can now proceed with a fresh copy of the latest source code for analysis, or (2) the reaction to an imminent or actual bankruptcy, merger, or acquisition of the software vendor. In this case the ownership of the source code would be transferred to the enterprise. But now some organizations may behave like the dog that chased the car – they are not prepared to do anything useful with the source code. Its developers, testers and architects are not familiar with that piece of code or the technology, there are no manuals or documentation available, or the development or staging environments will not support the source code, and the CIO may be expecting normal operations to resume at any moment.

The truth is, because the probability of the code escrow being triggered is (hopefully) low, it is only natural that an organization's readiness to use that same source code should also be quite low (as higher readiness equals higher costs). In such situations, it is better to manage expectations in a realistic manner, and not assume that just because you have the source code everything will go back to business as usual. It is likely that the source code will now be used as part of the disaster recovery or business continuity processes started by the sudden loss of a strategic partnership with a software vendor. Development resources will need to be diverted to the newly acquired source code, first to understand and maintain it, and later to keep its regular life cycle. The development and staging environments may be strained by the insertion of the new technology, and a surge in urgent or emergency changes may take place, until the source code maintenance is repatriated, or transferred to a new vendor. All these factors need to be considered when devising a strategy that makes use of code escrow services, or its purpose may be defeated by lack of preparation.

Who should use code escrow services?

It is not uncommon for larger organizations to use SaaS services. Salesforce.com, a leader in the SaaS environment, has clients spanning different industries including insurance, health care, communication, and several others.[8] This demonstrates that organizations of all sizes have accepted this business solution as standard practice. According to IronMountain,[9] 75% of Fortune 500 and 75% of FTSE use code escrow agreements, and another study claims that 80% of all Fortune 1,000 firms have at least one software package on deposit with an escrow agent.[10]

Amoco Oil Corporation provides a successful example of custom-made software being protected by code escrow. The energy corporation acquired a new technology that, although very promising, was developed by an unproven software provider. Amoco exercised diligence, and as part of the technology acquisition it included contractual controls wherein the provider accepted to escrow the source code and release in case of bankruptcy. Shortly after that the provider went out of business and Amoco retained the software which was critical for its operations.[11]

Code escrow is not a perfect solution for all companies and scenarios; however, there are certain risk areas in which it is definitely prudent to consider using it. Examples would include the transfer of website ownership, the use of third-

Please continue on page 44

7 "Source Code Escrow: Are You Just Following the Herd?" *CIO Magazine* – http://www.cio.com/article/187450/Source_Code_Escrow_Are_You_Just_Following_the_Herd_.

8 http://www.salesforce.com/customers.

9 IronMountain website http://www.ironmountain.co.uk/resource/datasheets/IPMOverviewUK.pdf.

10 W. D. Denson, "The Source Code Escrow: A Worthwhile or Worthless Investment?" – http://www.bankruptcy.rutgers.edu/source_code_escrow.pdf.

11 Ibid.

Scource code escrow continued from page 40

party software for critical functions (SaaS – some companies even specialize in escrow services for SaaS providers[12]), or outsourcing the creation of critical applications such as a critical financial system used by a bank and developed and maintained by an independent software development company. In such a scenario, the bank relies on the application to conduct transactions but does not have access to its source code, nor does it control the financial health of the development company or its internal resources and management practices. This situation exposes the bank to considerable risk. Whenever the scenario includes a critical piece of software and substantial risk involving its developer, code escrow should be considered.

Conclusion

Source code escrow is a non-intrusive and essential area of risk management that is often overlooked when protecting information systems. Without the original source code, it is challenging to conduct code reviews and assess risks that an application is subjected to, as well as investigating incidents related to fraud and misuse of applications.

Escrow service is a viable alternative that can be considered similar to insurance: sometimes it is never used. However, when used it can be the last line of defense when everything else fails. It is an inexpensive and simple control that can be used to ensure that the code your organization relies upon is available to you when the worst case scenario occurs.

About the Authors

Raoul Gomes is a security advisor currently working with Holding Trust Inc. He can be contacted at raoulg@gmail.com.

Rafael Etges is the National Practice Leader for Governance, Risk & Compliance at TELUS Security Solutions. He can be contacted at rnetges@yahoo.com.

The opinions expressed in this article do not necessarily reflect the views and policies followed by the author's employers.

12 http://www.nccgroup.us/software-escrow/saas-escrow.aspx, also "SaaS: Good for customers, vendors, or both?" – http://news.cnet.com/8301-13505_3-9892135-16.html.