Bringing Common Criteria certification to web services

Samuel Paul Kaluvuri
Applied Research Security&Trust
SAP Labs France
samuel.paul.kaluvuri@sap.com

Michele Bezzi
Applied Research Security&Trust
SAP Labs France
michele.bezzi@sap.com

Yves Roudier
Eurecom
Biot, France
yves.roudier@eurecom.fr

***Abstract*—Solutions based on service-oriented architecture are gaining popularity. However a wider adoption, especially for business critical functions, is hampered by the trust deficit that exists between consumers and providers, as consumers are shielded from the service architectures and the operation of the service itself. Security certification can be used as a means to bridge this trust deficit. Common Criteria for Information Technology Evaluation (CC) is a widely recognized and used security certification scheme. However, the CC scheme was tailored to provide assurance for traditional software provisioning models and hence cannot be applied to SOA solutions as is. In this paper, we present the limitations of the CC scheme when applied in SOA, the challenges that must be overcome for its adoption and possible directions through which some of those challenges can be met. In particular, we point out that CC scheme should be extended to allow for dynamic evaluation of deployed systems (which includes the operational environment) and for handling assurance of composite services.**

***Keywords*-Security Assurance; Security Ceritifcation; Web Services; Common Criteria;**

I. Introduction

A paradigm shift in software provisioning and consumption models is taking place due to the adoption of Service Oriented Architectures (SOA). Cloud solutions, based on SOA, such as Gmail, Dropbox, Saleforce, Successfactors enjoy a huge popularity among consumers. Such solutions offer enormous benefits to consumers by insulating them from the complexity and costs associated with procuring and maintaining the required IT infrastructure while providing a large scale inter organizational, interoperability. These characteristics of SOA raise concerns among consumers regarding the security of the service [6]. In fact, security concerns hamper a more wider adoption of service based solutions, especially in critical domains such as health-care, finance, defence, etc.

In order to address their apprehensions over the security of the service, consumers establish Service-Level Agreements (SLAs) with the service providers. However, typically, SLAs involve two or few predefined parties only, and this is not a scalable solution for a service landscape, which is very dynamic, and it can result in a "closed SOA" where only few entities are involved in the service provisioning and consumption. Though there has been work [11] in the direction of establishing dynamic SLAs, it is focused more on the quality aspects of the service rather than security. Security certification of services can constitute a valid solution for gaining security assurance in the dynamic service landscape, and yet preserving the "Open SOA" principle. In the certification approach, a trusted third party evaluates and, consequently, certifies that a service has certain security properties. Consumers can then rely on the "'stamp"' of approval of the certification authority, without the necessity to establish additional bilateral contracts with the specific service provider.

There are several existing security certification schemes for Information Technology (IT) products, and among them Common Criteria for Information Technology (CC) [4] is the most widely recognized and used security certification scheme. The CC scheme certifies products that range from software, firmware to hardware. A quick survey of CC certified products reveals that though there are more than 1500 products certified in various categories of products, such as Operating systems, Databases etc., however there are no services that are certified by the CC scheme [3]. This can be a direct consequence of a lack of interest from service providers to gain (and from consumers to demand) CC certification for services. The major reasons include: *a)* CC certification is an expensive process [10] and may not justify the Return on Investment (ROI) for service based solutions; *b)* CC certification is a lengthy process often taking upto 8-12 months to pass the evaluation [10] - a major limitation in service environments where services are constantly updated; *c)* CC scheme is tailored for traditional software provisioning models and does not cope well with service environment.

In this paper we present the issues that arise while adapting the CC scheme to service environments and provide possible directions through which those issues can be resolved. In the next Section, we will describe a service scenario, which will be used in the following to illustrate the challenges of certification for services. Section III will discuss the major challenges, and Section IV describes possible directions to address them.

II. Scenario

Let us consider the example of a popular cloud storage service Dropbox. The security policy of Dropbox service [5]

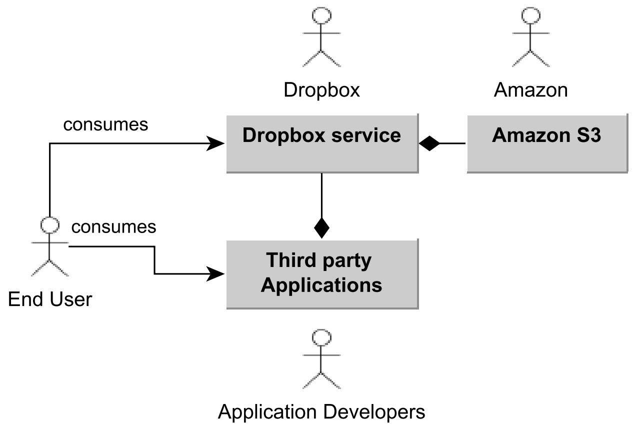

Figure 1. Dropbox Overview

discloses that it uses another cloud storage service, *Amazon S3*, to store its consumers' data. However, Dropbox ensures that it uses strong encryption mechanisms to secure its consumers' data before storing on the *Amazon S3* service. This ensures that consumers' data is kept confidential despite the data being stored externally.

Dropbox also exposes an API to third party applications that, with due prior authorization from the consumers, permits access to the consumer data on their Dropbox account. An example of such service can be an image processing service that accesses the consumer's photos stored in their Dropbox account and process those images (such as resizing, applying filters, and so on) and store them again in the Dropbox account of the consumer as shown in Figure 1. Such applications can be extensively found in the mobile domain (Google Play, Apple AppStore). These examples illustrate the ease of creating a chain of compositions in a SOA environment and certification of such services through the CC scheme needs to provide consumers with security assurance that allays their concerns stemming from the lack of control over the service and its operational environment.

III. Challenges

The CC scheme, in its present form, does not cope well with service environments and hence cannot be used to gain the necessary security assurance to consumers. There are two main reasons: First, in the CC evaluation process, only the chosen "deployable" configurations are evaluated and certified while making assumptions on the operational environment for the secure operation of the product. From CC evaluation perspective, these assumptions are always considered fulfilled, and the consumer is delegated the responsibility of operating the product in an environment that satisfies these assumptions. In fact, such strong assumptions on the operational environment are not viable in a service landscape as, very often, neither the service provider nor the consumer have any control over the operational environment. Moreover, certifying "deployable" configurations of a service does not provide much value to consumers, as they need assurance on the "deployed" service. Second, CC scheme is a system wide certification and is not devised to certify modular systems. Though the CC scheme proposes the usage of Composed Assurance Packages (CAP) to gain assurance over composed systems it does not provide high levels of assurance [9], and, as a matter of fact, these packages are not used. In the following subsections, we discuss in detail the issues that arise from these two key aspects.

A. Assurance of Deployed Systems

In Common Criteria certification the Target of Evaluation (*TOE*), which describes the parts of the product that are subject to evaluation, delegates responsibility to the Operational Environment (*OE*), which is the environment in which the *TOE* operates, for its secure operation. Typical examples can be Application Server or Database products that require the Operating System, which is part of the OE, to provide user role management. In traditional software provisioning models, these objectives for the *OE* serve as a guidance for the consumers to configure their *OE* to ensure the *TOE* operates securely. This is possible only when consumers have control over the *OE* which is the case in the traditional software provisioning models.

However, in service environments consumers do not have neither any control over the service (*TOE*) and its execution environment (*OE*) nor transparency regarding the service architecture. In such scenarios, certifying "deployable" configurations does not provide any meaningful assurance to service consumers. Only certifying the "deployed" services can provide the required assurance to consumers. Certification of a deployed service can be done similarly to the certification of a (set of) "deployable" configuration(s) of the service. But the service landscape is dynamic and neither the service consumer nor the certification authorities have control over the service and its operational environment. Thus, even when a service is certified, consumers cannot be certain that the service that is being consumed is operating in the certified configuration. The reason for this is due to the static certification lifecycle of the CC scheme, where once a product passes the evaluation and is certified, the role of the CC authorities end and the onus is on the consumer to ensure that the product he procures is the same as the one that is certified. The CC scheme's " Assurance Continuity" [2] allows incremental versions of a product to be certified by evaluating only the changes made to the products. However, this does not reflect any proactive role of the certification authorities, it merely provides software providers to reduce the time and expense to get newer versions of their certified products evaluated. Clearly, this *static* certification lifecycle does not scale to *dynamic* service environments.

B. Assurance of composed services

A key feature of SOA is the ease through which services can be composed to form complex, composed application as shown in the scenario presented in Section II. Indeed, interoperability of services is a major reason for the widespread adoption of service based solutions. Composition of services can happen at design time or at runtime and in both the scenarios, if we assume the participating services are CC certified, it is not trivial, (in the case of runtime compositions practically impossible) to gain the assurance of the overall composition. One reason being that composition was never an inherently addressed issue within the CC scheme, as it was designed to be a system wide certification. Another reason being the natural language representation of security certificates from the CC certification process.The CC scheme requires product vendors to disclose certain information regarding the system being certified, the security features implemented in the product, the assets being protected, the threats and so on, in a document called as the *Security Target (CC-ST)*. This document is seen as the descriptive part of the CC certification. The CC scheme only prescribes the content that must be captured in the CC-ST document, but does not prescribe any rules or structure for the representation of this content. The only standardized elements in the CC-ST document are the Security Functional Requirements (SFR) that the vendor claims the product meets and Security Assurance Requirements (SARS) that describes the rigour of evaluation of the product. These SFRs and SARs are prescribed in the CC standard, but in the context of the CC-ST they cannot provide complete information regarding the security of a product. This is a major limitation in performing any sort of automated reasoning on the security certificates, which is an essential step in facilitating certification composition to assess the overall assurance of a composed service.

IV. Possible Solutions

In this section, we present possible solutions through which some of the challenges mentioned before can be addressed. In particular we present some of the work we have done in order to address these challenges.

A. Assurance of Deployed Systems

Since current security certification schemes are unable to provide assurance of deployed systems, we propose several extensions to the existing state of the art. These extensions primarily change the scope of the evaluation and consequently the scope of the certification as well.

The CC scheme should allow for certification of deployed and running systems and keeping into account the dynamic landscape of services. This would require the scheme to evolve from a purely static certification lifecyle to a dynamic certification lifecycle. A dynamic certification lifecycle requires a more active role played by the certification authorities, which currently stops after issuing a certificate. Its role should be extended to involve "monitoring" of the deployed service by the certification authorities. In the CC scheme there are no such mechanisms prescribed, as the product operation falls under the purview of the consumer.

Hence we propose a *Dynamic Certification Lifecycle* for CC certified services, that enlarges the role of the certification authorities to include " service monitoring" once they are certified. Such monitoring mechanisms ensure that the service and its underlying *OE* have not changed from the certified configuration and, when any changes are detected by these monitors the certification authorities can flag the corresponding security certificates of those services. These *service monitors* must use the contents in the certificate to choose the aspects of a service (or its execution environment) to monitor. A key aspect that impacts the nature and scope of such monitors are the trust relations that exist between the different entities involved (i.e. certification authorities, service providers etc.,) [1].

In addition, the dynamic nature of service landscape poses a challenge to the current evaluation methodologies used by the certification authorities. Currently, evaluation is performed at a point in time, and does not suffice in service environments where the evaluation of certain aspects of the service can only be verified at runtime. For example, when considering a security feature such as available of data, static evaluation alone cannot provide the required assurance. In fact at runtime, there might be several factors that can affect this property. Another example is the Dropbox service, presented before, which uses Amazon S3 service to store data - but consumers might have a preference that their data must not be stored outside the EU region, and such properties can only be verfied at runtime and not statically.

Accordingly, the certification processes should extend to encompass dynamic evaluation of these aspects of services [12]. A key challenge to overcome, is the identification of aspects that could be verified statically from the ones that should be verified dynamically, as well as to determine the relevant information that should be captured in the certificates to perform this dynamic evaluation. For example, in addition to the information regarding the results of the static evaluation performed on the service, the security certificates could be enriched with information to enable the execution of test-suites at runtime.

B. Assurance of Composed Services

The CC certification should address the issue of certifying composed services. A key step forward in this direction is to move towards a digital representation of the security certificate, which is the result of the CC certification process, from the current natural language representation.

We recently proposed the concept of *Digital Security Certificate* [7], that provides a structured representation of the contents in the CC-ST. This structured representation of the security certificate is realized by a language [8] that is used to produce machine readable security certificate artefacts that are encapsulated as SAML assertions. The resulting artefact which we refer as ASSERT allows the usage of the security certificate of services in typical SOA scenarios such as service discovery, service selection, service composition to name a few.

This structured representation takes into account the composability of services and so it is designed in a manner that facilitates the composition of certificates by representing the various elements in the certificate in a fine grained manner and providing explicit links between them. This is in stark contrast to the CC-ST where the assets, threats and various other elements that are described, have implicit links only. This may be sufficient for documents that are aimed for human consumption, but not suitable for automated processing of the security certificate.

The key elements in the ASSERT artefact that are relevant for the CC scheme are: Service Description, Security Property Specification, User Defined Extensions. These three elements are used to provide consumers with information about the security aspects of the service at varying levels of abstraction. The Service Description *(SD)* element provides information about the service and its underlying architecture, there by mitigating the concerns of consumers on the lack of transparency of services. The Security Property Description *(SPS)* provides details about the security properties of the service at varying levels of abstraction while the User Defined Extensions *(UDE)* can be used by certification authorities or service providers to provide additional information to consumers. In a CC-ST, the *Target of Evaluation(TOE)* identifies the system that is being certified and the boundaries of the evaluation are indicated in an natural language. But for a machine readable certificate there should be a clear distinction between the system that is being certified and the aspects of the system that are subject to evaluation. And it is not enough to describe just the system that is being evaluated as it is of utmost importance in service based systems, due to the fact that services can be easily composed of external services and should be a part of the service description where as these external services wont be subject to evaluation. In addition, the *TOE* in the CC-ST contains the system architecture, the different components that compose the system among other information such as configuration in which the system is evaluated, the underlying IT architecture etc., and this is represented in natural language accompanied by architecture diagrams. This is another aspect that needed to be considered for a machine processable *TOE* description. So in order to address these two issues, we introduced an element called *Target of Certification (TOC)*, which is composed of *TOC Components* and a *Deployment and Implementation Model* that explains how these components are composed. Another key element of any security certification scheme is identification and description of assets that need to be secured. In CC-ST, the assets are described in natural language and no identifiers are provided for these assets and hence an explicit link cannot be provided between the security properties and the assets that they secure. In the ASSERT language, we adopt a asset-centric approach and the assets are clearly identified and described in the *SD* and referred in the *SPS*. Each *TOC Component* contains a set of assets that need to be secured along with a component model describing the internals dynamics of the component at a certain level of abstraction as deemed sufficient by the certification authority, given that service providers might not want to disclose proprietary information. Also some aspects of the component have no relevance on the security property that is certified and hence such aspects can be abstracted away from the component model. The *TOE* is part of the TOC, referring to a subset of the *TOC components* that will be subject to evaluation.

While the ASSERT language prescribes the structure on the representation, the actual content of the certificates should come from an ontology that is prescribed by the CC scheme. The language facilitates such integration with external ontologies, keeping into account that the certification criteria evolves over time as well as to accommodate any variations between the different country specific variations of the CC schemes (i.e., US, UK etc.,) which is the case for products certified at higher levels of assurance.

These machine processable certificates facilitate advanced and automated reasoning, thus providing more concrete assurance to consumers. For example, if we consider the scenario presented before in Section II, where the Dropbox service uses the Amazon S3 service, it delegates some responsibility to Amazon such as redundant data storage for availability of the data. However, currently Dropbox's security policy only points to the security policy of Amazon S3, but it still requires human inspection to verify. Potentially, having available the ASSERTs of Dropbox and Amazon S3, the consumer could verify whether the security features delegated to Amazon S3 by Dropbox (such as redundant data storage for availability of data) are actually met by Amazon S3's security certificate. This would require additional tool support, but given that the ASSERTs are machine processable such verifications can be performed with relative ease.

V. Conclusion and Future Work

In this paper we discussed several important issues that prevent the application of CC certification to the SOA domain. We argued that the CC scheme needs to adapt to the new software provisioning models where security assurance needs to be provided in a modular manner that scales to the dynamic nature of service landscape. We discussed key challenges that need to be addressed, in particular providing assurance for deployed systems and composed services as opposed to the current practice of providing assurance to deployable and standalone systems. We present possible solutions through which these issues can be addressed by proposing a Dynamic certification lifecycle as well as a Dynamic evaluation process. We present a concept of Digital security certificate, which allows automated reasoning to be performed on them to gain assurance over composed services.

Acknowledgment

This work was partly supported by the EU-funded project ASSERT4SOA (grant no. 257361).

References

[1] M. Bezzi, S. P. Kaluvuri, and A. Sabetta. Ensuring trust in service consumption through security certification. In *Proceedings of the International Workshop on Quality Assurance for Service-Based Applications*, pages 40–43. ACM, 2011.

[2] Common Criteria. Common Criteria Assurance Continuity:CCRA Requirements, 2004.

[3] Common Criteria. Common criteria: Certified products list - statistics, 2012.

[4] Common Criteria. Common Criteria Part 1: introduction and general model, 2012.

[5] Dropbox inc. Dropbox security overview. http://www.dropbox.com/dmca, accessed on 3-3-2013, 2013.

[6] Gartner-Report. Demand for Cloud-Based Offerings Impacts Security Service Spending Eric Ahlm, April 2013.

[7] S. P. Kaluvuri, H. Koshutanski, F. Di Cerbo, and A. Mana. Security assurance of services through digital security certificates. In *Web Services (ICWS), 2013 IEEE 20th International Conference on*. IEEE, 2013.

[8] H. Koshutanski, A. Maña, R. Harjani, M. Montenegro, S. P. Kaluvuri, F. Di Cerbo, E. Damiani, C. A. Ardagna, M. Anisetti, D. Presenza, S. Gürgens, R. Menicocci, V. Bagini, F. Guida, and A. Riccardi. *ASSERT Language V2*. http://www.assert4soa.eu/deliverable/D1.2.pdf, 2012.

[9] M. Maidl, D. von Oheimb, P. Hartmann, and R. Robinson. Formal security analysis of electronic software distribution systems. In *Computer Safety, Reliability, and Security*, pages 415–428. Springer, 2008.

[10] U. S. G. A. Office. Information assurance: National partnership offers benefits, but faces considerable challenges. Technical Report GAO 06-392, Report, March 2006.

[11] H. Rasheed, A. Rumpl, O. Wäldrich, and W. Ziegler. A standards-based approach for negotiating service qos with cloud infrastructure providers. In *eChallenges Conference*, 2012.

[12] G. Spanoudakis, E. Damiani, and A. Mana. Certifying services in cloud: The case for a hybrid, incremental and multilayer approach. In *High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on*, pages 175–176, 2012.