Trusted Web Service

Zhexuan Song, Sung Lee, Ryusuke Masuoka

Fujitsu Laboratories of America, Inc.

8400 Baltimore Avenue, Suite 302

College Park, Maryland 20740, USA

{zhexuan.song, sung.lee, ryusuke.masuoka}@us.fujitsu.com

Abstract

"Remote Attestation", a concept well known in the Web Security domain, provides the capability to measure and verify the hardware and software settings of the platform where a Web Service request originated, thus greatly improving the Web Service security. However, in many Web Service security implementations, remote attestation is not supported.

In this paper, we will introduce an ongoing project called "Trusted Web Service" (TWS). TWS adopts a two-tier model. At the lower tier, TWS uses the Trusted Network Connect (TNC) standards to measure, encrypt, and transfer the measurement data (such as the hardware/software states of the platform) from a client to the server, and blocks potentially malicious clients from accessing the network. At the upper tier, TWS extends the OASIS standards and uses the aforementioned measurements in checking against predefined policies in order to provide more advanced control. TWS leverages existing standards whenever possible and suggests a better security model for Web Services.

1   Introduction

Web Service implementations support interoperable machine-to-machine interaction over a network. Applications that are written in different programming languages and running on various platforms can use Web Services to exchange data over a computer network, due to its inherent support for interoperability. The key factor to achieving interoperability is the use of open standards (such as WSDL[6], SOAP[4]); thus, Web Services are becoming the foundation of Service Oriented Architecture (SOA).

SOA expresses a new software architecture, which uses services to fulfill the user requirements. Recently, SOA has gained a broader industry acceptance as Web Service standards became more popular. However, some obstacles still remain. One of the major concerns is the security of Web Services. It is a valid and reasonable concern as the practice has been that security issues were not carefully studied and integrated in the design stage of Web Service standards. The existing "Web Service security solutions" are rather add-on modules to the already-defined Web Service standards.

The existing Web Service security solutions (such as Web Service Security [7] by OASIS[2]) authenticate Web Service requests based on the information provided by clients. Information supplied by clients varies depending on particular protocols being used. Examples of such information are username/password combination, web certificate, security token, bio-metric reading, and so on. However, these solutions have one common drawback. Since the information is provided by the Web Service requester, it is hard for the service provider to distinguish the information coming

from the "right" source vs. a malicious source. For instance, let us suppose a Web Service that requires client-side certificates to authenticate requests. If a hacker steals the certificate from a legal user's computer and starts to use it, he will receive the same level of service as the legal user would. The Web Service provider has very limited capabilities to identify and isolate the service requests coming from a compromised client.

Yoshihama, et. al. [9] have attempted to address this problem. They have identified an important factor that should be integrated in the authentication process: "Remote Attestation". Remote attestation provides the ability to measure and remotely verify the hardware and software components of the platform from which the Web Service request originated (remote in most cases). It establishes "trust" among distributed parties. Their solution is called "WS-Attestation", where remote attestation is accomplished using Trusted Platform Module (TPM) technology.

While it is important to remotely measure and verify the integrity of hardware and software components of the Web Service requester, their solution would be insufficient in two points. First, some of their work is overlapping with existing TCG [5] standards, or more specifically, Trusted Network Connect (TNC) standards. As TNC standards have been designed to be interoperable and comparable with existing solutions, they are more likely be adopted by companies and vendors. Second, in their WS-attestation proposal, all attestation validation happens at the Web Service layer, which inevitably degrades the performance of the Web Service and eventually leads to scalability issues.

In this paper, we introduce the Trusted Web Service (TWS). Our design principle is to utilize existing standards whenever appropriate and not to re-invent the wheel. We realized that TNC offers a good infrastructure for creating secure network connections based on measurement metrics collected on the client's platform. Since the validation occurs at the network layer, it will not affect the performance of applications built on top of it, such as Web Services. On top of the TNC layer, we deploy Web Service and OASIS security suites (see Section 2.1 for details). Please note that the metrics collected on client's platform and used at the TNC layer are also available to the Web Service. This makes it possible to define advanced policies at the Web Service layer. TWS is simple and mostly based on existing open standards; therefore, it is more likely to be accepted.

The paper is organized as follows. In Section 2, we will give a brief introduction of related work, namely OASIS suites and TNC. Then, in Section 3, we outline the architecture and the workflow of TWS. One application scenario is discussed in Section 4; the last section concludes the work.

2 Related Work

In this section, we will introduce two major components that are involved in TWS.

2.1 OASIS Suite

OASIS (Organization for the Advancement of Structured Information Standards) security committee develops a set of security standards for Web Services, including WS-Security [7], XACML [8], SAML [3], and so on.

WS-Security extends SOAP with a general-purpose mechanism to associate security information with SOAP message content. It also suggests a framework that encodes binary security tokens and includes encrypted keys. SAML is an XML-based language for creating and exchanging user authentication, entitlement, and attribute information between a client and the server. XACML is another XML-based language for defining and evaluating access control policies (mainly used on

the server side). The suite (WS-Security, SAML and XACML), along with other OASIS standards covers the complete workflow of the Web Service invocation process.

As we discussed before, the missing part in the OASIS suite is remote attestation. The information that the server requires is mainly provided by a client. Thus, the Web Service server has no confidence on whether the request is from the authenticated user or from someone who has stolen the legal user's identity. Furthermore, if the malicious software is installed on the user's computing platform without permission, the OASIS suite can not detect the potential security hole and prevent the attack.

Therefore, we believe that remote attestation must be included into the system to better protect the integrity of both the client and the Web Service server.

2.2   TNC

Trusted Network Connect (TNC), an initiative of TCG, addresses and attempts to provide network access control that meets security requirements through open-source, non-proprietary standards. In order to ensure interoperability and compatibility with the existing network infrastructure, the TNC architecture is designed to utilize existing industry standards and protocols such as Extensible Authentication Protocol, Transport Layer Security, and RADIUS. The TNC architecture supports commonly used access mechanisms such as VPN, SSL, dial-up remote access and other networking technologies including wired and wireless networks and 802.1x infrastructure.
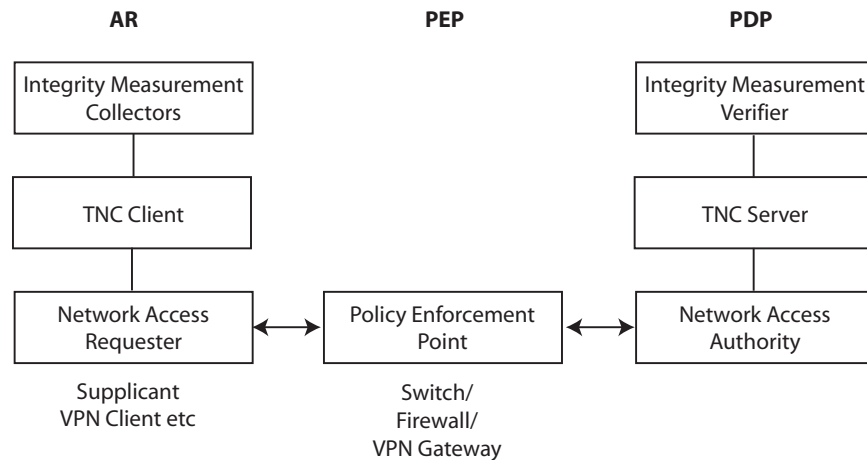

Figure 1: TNC Architecture: This diagram is based on the TNC Architecture Specification

There are three entities in the TNC architecture (Figure 1): an Access Requester (AR), a Policy Enforcement Point (PEP), and a Policy Decision Point (PDP). When an AR makes an attempt to access a protected network behind PEP, the access request is passed through an integrity evaluation process in order to determine what level of access should be granted, if any. Based on the measurements collected at the AR and the policy configuration, the PDP makes a decision. The PEP, usually a network access device such as a switch or wireless access point, enforces the decision made by PDP by granting the appropriate access to the AR. The TNC architecture allows for an option to include platform measures such as Platform Configuration Register (PCR) values from Trusted Platform Module (TPM) of the AR.

We believe that TNC standards offer a network access control solution with interoperability and compatibility with existing networking infrastructure and they will be easily adopted by many

networking vendors.

3 Trusted Web Service (TWS)

This section is an outline of Trusted Web Service (TWS). The architecture of TWS is depicted in Figure 2.
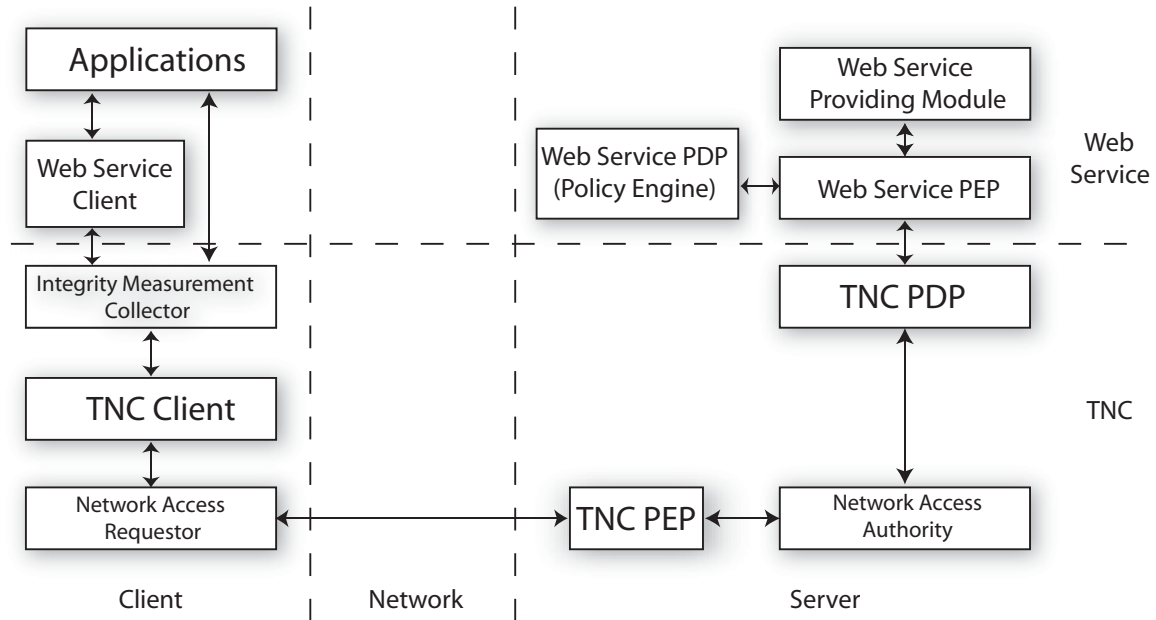

Figure 2: Trusted Web Service architecture

TWS architecture includes two layers. The lower layer is the "TNC Layer" and the upper layer is the "Web Service Layer". The TNC layer implements the Trusted Computing TNC standards. The TNC client could be an 802.1x user, a VPN client, or a Web browser initiating SSL connections. When a client from an unprotected network attempts to access resources in a protected network, it must go through an "Integrity Check Handshake" process. The TNC PEP consults the TNC PDP to determine what level of access should be granted, if any. Integrity measurements collected at the TNC client are passed to and verified at the TNC PDP to grant access based on previously configured policies.

The Web Service layer implements Web Service standards (such as SOAP) and OASIS suites. On the server side, the "Web Service PEP" assigns each Web Service request an "action level". Two action levels (accept / reject) are mandatory, but more levels in-between are also possible. The Web Service PEP is supported by the "Web Service PDP" (policy engine). Web Service PDP stores a set of polices defined in advance and makes decisions based on these policies. While defining a policy, both the user's identity and the attestation measurements are considered. One policy example is "the request is rejected (action level = reject) if no anti-spyware is detected in the requester's platform". We expect to use XACML to define these policies.

We separate the Web Service PEP from the TNC PEP for two reasons. First, the TNC PEP is the enforcement point for all network access, while the Web Service PEP is specifically targeted to each Web Service. Within the same network, multiple Web Services might be provided. If all decisions are made at the TNC PEP, the policies that are involved in each Web Service have to be

put together, which complicates the system and deteriorates the performance. Second, for different Web Services contradictory rules might be defined and it would be hard to configure a TNC PEP that enforces all these rules. In general, we strongly believe that the TNC PEP should enforce rules that are only related to network access, and advanced Web Service related rules should be defined and imposed by each Web Service PEP.

The typical work flow is as follows. The access to Web Services will be determined in two phases. During the first phase, the requester goes through a typical TNC integrity check handshake process in order to gain access to the protected network. Starting from a client, applications or Web Service clients issue a Web Service request. At the end of the request, the Integrity measurement collector attaches the measurement optionally sealed and sent with related PCR values. The measurement is checked and the TNC PEP accepts/rejects/isolates the request based on the rules defined in the TNC PDP.

Once the general network access is granted following the TNC standards, the Web Service request, along with measurements used by the TNC components, is given to the Web Service server. Within the server, the Web Service PDP will check the request along with the measurements and determine the action level. Based on the action level, the Web Service PEP will either send an "access denied" exception, or transfer the request to the Web Service Providing Module along with the calculated action level. Once the access control process is done, the remaining interactions are just like what we have today.

We are in the process of building a TNC test bed, which includes all TNC modules, comprised of products from different vendors and open source components. In parallel with this effort, we are studying the OASIS suites. Our main focus is to understand how to extend the OASIS standards and allow platform attestation data to be included in messages. Finally, we plan to implement the complete TWS workflow and apply the idea into real-life applications. One possible application domain is discussed in the next section.

4 Application

A medical record retrieval system is a possible candidate for this technology. According to the HIPAA (Health Insurance Portability and Accountability Act), health and medical information is private and should be protected; US federal laws give patients rights over their own health information while setting rules and limitations on how others can access it. The information can only be shared for the purpose of treatment, health care, and care coordination [1]. The benefits of having electronic medical records systems have been acknowledged for some time. However, stringent requirements and regulations such as HIPAA prevented electronic medical records systems from being widely used. As a Web Service that provides medical records, these requirements must also be met.

We use an example to explain the requirements. Suppose a patient grants a dentist the permission to retrieve her health report for treatment purpose. She agrees to the following: 1) the dentist can retrieve her information only in the office from a registered machine and the latest version of anti-spyware must be installed in order to prevent possible information leak. 2) the dentist may only retrieve her general information and dental records; any other sensitive data, such as her psychiatric evaluation report, should not be available to the dentist.

TWS is the best technology for this kind of applications. While the doctor is requesting the medical information of a patient, the doctor's computing platform measurement is collected. At the TNC layer, the basic checks are performed, such as whether the platform is registered beforehand

to a doctor that is involved in the program, whether any anti-spyware software is installed, etc. After the request passes the TNC check, it will be sent to the Web Service layer where another round of checks happens, such as whether the correct user login information or user token is provided. Finally, after the the verification is completed, the Web Service Providing Module will send a report, which includes only the information relevant to the treatment. The TNC standards provide protection of the data while in transit through various mechanisms offered during message transportation.

5 Conclusion

In this paper, we discussed the importance of remote attestation in building a secure Web Service and introduced Trusted Web Service (TWS). Our proposal combines the OASIS Web Service standards with the Trusted Computing TNC standards. The TNC layer of TWS collects the measurement metrics on the client's computing platform and determines the network access (*allow, reject, or isolate*) based on previously defined policies (within TNC PDP). The Web Service layer uses not only the authentication information provided by the user (as regulated in OASIS standards), but also the measurement metrics collected by the TNC layer in order to establish (within the Web Service PDP) 1) whether or not the service request should be accepted under the current client's platform settings, and 2) at which action level the service should be provided.

The work presented here is still at a conceptual level, and we are in the process of following through our development plan and implementing TWS. Once we verify our concepts, then we plan to apply this technology to real-life applications, such as medical record retrieval systems as described. Additionally, we are looking into other mechanisms to enhance authentication and their integration into TWS.

References

[1] HHS – office for civil rights - HIPAA. http://www.hhs.gov/ocr/hipaa/.

[2] Organization for the Advancement of Structured Information Standards. http://www.oasis-open.org/home/index.php.

[3] Oasis security services (SAML) TC. http://www.oasis-open.org/committees/tc_home.php?-wg_abbrev=security.

[4] Simiple object access protocol (SOAP). http://www.w3.org/TR/soap/.

[5] Trusted computing group. http://www.trustedcomputinggroup.org/.

[6] Web service definition language (WSDL). http://www.w3.org/TR/wsdl.

[7] Web service security. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss.

[8] Oasis eXtensible Access Control Markup Language (XACML) TC. http://www.oasis-open.org/-committees/tc_home.php?wg_abbrev=xacml.

[9] Sachiko Yoshihama, Tim Ebringer, Megumi Nakamura, Seiji Munetoh, and Hiroshi Maruyama. Ws-attestation: Efficient and fine-grained remote attestation on web services. *ICWS*, 0:743–750, 2005.