Service Discovery: Threats and Solutions

Slim TRABELSI[*], Yves Roudier[*], and Jean-Christophe Pazzaglia[+]

[*]*Institut Eurécom 2229 route des Crêtes, BP 193, 06904 Sophia-Antipolis, France*
E-Mail : {slim.trabelsi,yves.roudier}@eurecom.fr

[+]*SAP Labs France 805 Avenue du Dr Donat - BP 1216, 06254 Mougins Cedex – France*
E-Mail : Jean-Christophe.Pazzaglia@sap.com

1. Introduction

The Service Oriented Architecture (SOA) programming paradigm, currently promoted by Web Services, features a loosely coupled interaction model. Protocols and procedures enable the efficient interconnection of application subsystems or software components, through their service interface. Web Services overstep the limitations of traditional SOA solutions like CORBA and Jini in that they increase the dynamicity and flexibility of distributed software with XML-based interfaces, of which WSDL is a perfect example. Service discovery is an essential part of service orchestration in that it allows the dynamic detection of services available in a network. Numerous Web Service discovery solutions like UDDI, WS-Discovery, or OWL-S based discovery have been proposed in recent years although they do not address most security and trust issues. In WS-Discovery, for instance, security is limited to the non-repudiation, integrity, and freshness of messages. This is not enough to protect sensitive information about services from becoming available to rogue users, for instance. Uncontrolled data exchanged during a service location may threaten user privacy and citizen rights in the sense of the European Directive 1995/46/EC[1]. Healthcare scenarios provide a good example of such protection requirements for patient data [9]. This paper details threats to service discovery and how WS-Discovery may be extended to incorporate appropriate confidentiality and privacy protections.

2. Service Discovery Threats

Securing ubiquitous services means disseminating discovery information without exposing user or service information to their potentially hostile environment. Still, the description of threats to service discovery has not been addressed before [1] and [2], which clearly illustrates the lack of security in service discovery protocols. This section introduces a more detailed threat model attached to service discovery mechanisms and in particular which parts of this mechanism would be worthy targets to adversaries. This section provides a non exhaustive list of threats and the possible attacks that can be built against the data and resources of service discovery players (client, service, registry).

[1] *European Parliament and Council of Europe of Europe Directive 1995/46/EC, on the protection of individuals with regard to the processing of personal data and on the free movement of such data,* Oct. 24, 1995

Protocol Messages and Entities

- Denial of Service attack against the registry: the registry is not available (service-side): the attacker performs a Denial of Service attack by flooding registration messages. He intends to force the registry to consume its resources such that it can no longer provide its intended service
- Client request disclosure (client-side): client intentions, or activity, or identity may be revealed, directly or indirectly by his service lookup queries.
- Interception of request (client-side): the discovery request reveals private information about service discovery clients. A possible attack consists in faking the identity of a registry that is known and trusted and forwarding to that registry.
- Message modification or drop (client side): if the attacker compromised router from the network, he can intercept and modify or drop the client's lookup message to the registry
- Replay of lookup message DoS (client-side): the attack consists in replaying a lookup message coming from a legitimate client
- Replay of registration message (registry-side): the attacker replays the registration message of a properly authenticated service in order to update the service profile with wrong information

Service Registration (centralized architecture only)

- Registration to a malicious registry (server-side): an attacker might fake being a registry whose identity (and implicitly matching behavior) is known and trusted. Subsequent attacks include preventing clients to match the registered service for instance
- A service can be deregistered by an unauthorized party (service-side): occurs when an attacker tries to dereference an active service from the registry which it registered to previously
- Wrong registration (registry-side): An attacker can send a fake registration message to the registry containing wrong information with fake attributes.

Matching Process

- Client lookup disclosure (client-side): client intentions or activity might be disclosed if the matching process is open to all services registered. A service may have been established with the objective of gathering statistics about users trying to access a certain profile of services. More dangerously, an attacker might try to get access to confidential information sent by the client at the access phase subsequent to service discovery.
- Service discovered by unauthorized party (service-side): a typical example of this threat is the possibility for an attacker to determine the identity or content served by a service which wants to be seen or accessible only by a restricted set of other services (service trapping).

3. Architectures for Secure Service Discovery

This section studies two different approaches to securing service discovery, namely infrastructure based trust establishment, generally used with a centralized discovery scheme, and ad-hoc trust establishment, generally required to handle decentralized discovery.

3.1. Infrastructure Based Trust Establishment

The first approach to securing service discovery was to rely on an infrastructure for establishing the trustworthiness of clients and services. In the work of Zhu et al. [3], each participant to the discovery protocol is located behind a trusted proxy that sets up trust relationships through key exchanges with other proxies, while discovery is done through a normal registry. [4] suggests instead the use of a central entity combining the roles of a CA and registry, and helping clients and servers to set up a trust relationship and secure channels between each another.

Protecting Privacy during discovery has not been much addressed however. The threat model exposed in the previous section makes it clear that clients should be able to find a service matching their preferences, functionally as well as in terms of security and privacy requirements. The user should be sure that only services matching his preferences would be returned and trust in a service goes way beyond the simple authentication of its provider to encompass a complete certification of the service attributes. Symmetrically, the server does not know users, and should therefore be accessible only to client they trust to access them according to a precise behavior, which can only be guaranteed by some authority. Enforcing the verification of such certifications is therefore critical to service discovery. This task can be assigned to a trusted entity of the system. Whereas a new entity might take over this task, we rather suggest assigning it to the registry since the enforcement is simultaneous to service matching in such an architecture and since matching is already a trusted operation.

In [5], the registry has the role of a TTP filtering all messages according to Bloom filter based membership tests. In contrast, we settled for a much more expressive approach, which was implemented as an extension of WS-Discovery within the European project MOSQUITO[2]. This solution relies on the definition of XACML based discovery policies which may even be context-aware [6], made of rules specifying who can access the attributes in a client or service profile. These policies aim at (1) access control: discovery constitutes a preliminary form of access control to services by restricting the clients which will be able to subsequently contact a service. The sensitive resource here is the service's profile that must be hidden to the non authorized users. (2) privacy protection: the client can protect private information he reveals (identity, intentions, favorite services …) for each lookup performed from an uncontrolled disclosure. Discovery messages (publish and lookup) should contain credentials (certificate, key, or token) for the registry to authenticate their author and by a discovery policy to let the same registry know and enforce the participants' discovery preferences, the whole being signed.

3.2. Ad-Hoc Trust Establishment

The trusted registry based approach outlined above provides an efficient solution for fulfilling most security requirements of clients and services, yet it is bound to be deployed only where trusted registries are available (e.g., smart buildings). In contrast with this solution, trust may need to be established in an ad-hoc fashion. This is for example the case with the WS-Discovery protocol, which makes it easy to perform service discovery on top of a LAN or WLAN in that messages are multicasted to nearby clients and services. [7] described how attribute based encryption [8] can be used to protect sensitive information contained within WS-Discovery's *Discovery* messages. In such an encryption scheme, keys are obtained based on the attributes describing a party (role, identity, domain …). A client

[2] *Mobile Workers Secure Business Applications in Ubiquitous Environments (MOSQUITO) Project IST* 004636 https://www.mosquito-online.org/

may for instance specify which participants can read its requests by simply encrypting them based on the attributes he expects from these participants: only entities with the corresponding profile will then be able to decrypt messages. For example, if a client performs a lookup by sending $Encrypt[Request]_{\{storage,floor2\}}$ only the services that hold the private keys corresponding to a storage function and located on floor 2 will be able to decrypt and process the Request message. Similarly to PKI based approaches, these private keys would be provided by a trusted Public Key Generator (PKG) responsible for certifying services. However, once this certification is done, this PKG does not need to be contacted anymore. The service response may be protected quite symmetrically. The protection granted by this scheme is tightly dependant on the precision of the profile and how it captures client and service attributes.

4. Conclusion

This paper introduces a threat model of service discovery and presents two service discovery architectures that address part of these threats. In the first solution, with its centralized architecture, a trusted third party plays the role of a classical service registry and also enforces the discovery policy requested by users and services. The second one, with its decentralized architecture, relies on attribute based encryption to protect requests and responses and provides a flexible and decentralized access control functionality for limiting the discovery of private attributes to trusted users. The latter better addresses ubiquitous computing scenarios in which many services surround the user and also provides basic building blocks for the pervasive workflow model and its security extensions.

References

[1] S. Trabelsi, J.C. Pazzaglia and Y. Roudier *"Enabling Secure Discovery in a Pervasive Environment" 3rd International Conference on Security in Pervasive Computing* (SPC 2006) – York – UK – April 2006

[2] A. Leung and C. J. Mitchell, *"A service discovery threat model for ad hoc networks", in Proceedings of the International Conference on Security and Cryptography* (SECRYPT 2006), Setubal, Portugal, August 7-10, 2006, INSTICC Press, 2006, pp.167-174.

[3] F. Zhu, M. Mutka, L. Ni, *"Facilitating Secure Ad-Hoc Service Discovery in Public Environments", In Proceedings of the 27th IEEE Computer Software and Applications Conference*, Dallas, Texas, USA, 2003

[4] S.E. Czerwinski et al, *"An Architecture for a Secure Service Discovery Service", In Proceedings of MobiCom '99*, Seattle, WA, August 1999

[5] F. Zhu, M. Mutka, L. Ni *"Prudent exposure: A private and user centric service discovery protocol" Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Communications* (PerCom'04) Orlando, USA, 2004

[6] S. Trabelsi, L. Gomez, Y. Roudier *"Context-aware security policy for the service discovery" SNDS'07, 3rd IEEE International Symposium on Security in Networks and Distributed Systems*, May 21-23, 2007, Niagara Falls, Canada.

[7] S. Trabelsi, J.C Pazzaglia, Y. Roudier *"Secure Web service discovery: overcoming challenges of ubiquitous computing"* ECOWS 2006, *4th IEEE European Conference on Web Services*, Zurich - Switzerland, December, 2006

[8] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters *"Secure attribute-based systems". In CCS '06: Proceedings of the 13th ACM Conference on Computer and Communications Security,* Alexandria, Virginia, USA, 2006

[9] S. Trabelsi, Y. Roudier and J.C. Pazzaglia. "Service Discovery: Reviewing Threats and Security Architectures". Eurecom research report. May 2007. RR-07-197.