Quality of Protection Determination for Web Services *

Artsiom Yautsiukhin
University of Trento
evtiukhi@dit.unitn.it

ABSTRACT

Security is a very important aspect for Web Service technology. There are a large number of works devoted to security of Web Service transactions. However, we argue that security must be guaranteed for data processing (after transmission) as well. These requirements must be negotiated with a client and inserted into the agreement between a client and a contractor. The problem is that a client and a contractor have different views on how these requirements should look like. We propose a methodology which binds these views and describes a process for selection the security configuration that helps to achieve negotiated level of protection.

1. INTRODUCTION

Web Services is a rapidly emerging technology which has been developed to simplify business-to-business integration. It has a great potential to facilitate IT business outsourcing, when processing of an IT work package is delegated to an external organization. One of the important issues for Web Services is to shift relationships between involved parties to contractual ones. The first step in this direction is an unambiguous and clear definition of a *Service Level Agreement (SLA)* between a client and a contractor reflecting desired *Quality of Service* (QoS) (e.g. performance, maintenance). For this purpose XML-based specifications WS-Agreement [1] and SLAng [12] providing templates to describe QoS were proposed.

We would like to focus reader's attention on security requirements which should be inserted in the agreement. According to established standards (WS-Security [2], WS-Security Policy [6]), security requirements for Web Services are specified as policies which must be fulfilled in order to get access to the service. WS security standards do not mention data protection after transmission. The data may be corrupted during processing on contractor's server because of careless security management (e.g. data can be stored in a server without a properly configured antivirus). We argue that SLA must be extended with the section of an agreement that contains security requirements, which is called *Protection Level Agreement* (PLA). Similarly to QoS, we define *Quality of Protection* (QoP) as a set of security requirements a PLA guarantees. For more details we refer the reader to our previous work [10].

In this paper we provide a methodology for the aggregation of security requirements. It helps to select the most suitable security configuration according to a contractor's business process and different levels of trust between involved partners. The proposed methodology captures and binds security requirements useful for contractors with ones understandable by clients. Supported by a reasoning algorithm the methodology will be able to evaluate possible security system configurations. It will allow the contractor easily recalculate his QoP if a partner or his trust level has been changed or small system reconfigurations made.

The paper is organized as follows. In Section 2 we define a problem which emerges because a client and a contractor have different viewpoints on PLA. In Section 3 we propose our methodology where we: provide a strategy for QoP hypergraph contraction (Subsections 3.1), define a propagation function for the hypergraph (Subsection 3.2), decompose security services and link them with QoP hypergraph (Subsection 3.3) and briefly discuss how the algorithm for root QoP calculation should be implemented (Subsection 3.4). In the last section conclusions and future work are outlined.

2. PROBLEM

The crucial point in PLA negotiation is the identification of metrics which describe the level of protection. We have found useful to divide all metrics into two types:

- *Internal metrics* describe security qualities used by a contractor to achieve a high level of security.
- *External metrics* are negotiated with the client to show that her security requirements are addressed.

Some examples of internal metrics are: time between updates, length of passwords, percentage of compliance with a standard [7]. Possible examples of external metrics are number of successful attacks on client's data confidentiality [4] and mean time to intrusion affecting client's data [13].

*This work was partly supported by the project EU-IST-IP-SERENITY, contract N 27587

The main problem is that internal metrics are not informative enough for a client because they do not state explicitly how her assets will be affected by breaches in contractor's security system. On the other hand, external metrics do not tell the contractor how he should configure his system to achieve the metrics. The contractor must map the external metrics negotiated with client (PLA) to a functional security SLA to receive concrete requirements for security system configuration. In a sequel we will call the functional security requirements as *Qualities of Security Service (QoSS)*.

3. BINDING METHODOLOGY

We propose a methodology which helps a contractor to determine a QoSS satisfying the PLA negotiated with a client. In our methodology we use directed hypergraphs to capture structure of contractor's business process which determines how security requirements are distributed among its activities. A directed hypergraph is a generalization of directed graph where edges (or hyperedges) start from a *set* of nodes (source nodes) end end at a single node (target node) [3].

In our methodology we assume that a contractor and a client have negotiated a PLA using external metrics. We also assume that a contractor has a business process (BP) written in a hierarchical way. In other words, a provider defines a high level (abstract) BP (BP_h) where all activities are connected with one structural pattern (i.e. "sequence", "switch", "while", "flow"). Then for each non-atomic activity A_i a BP (BP_{A_i}) is determined. The decomposition continues until atomic activities are reached.

3.1 Phase 1. Build a QoP hypergraph

In the first phase of our methodology a contractor breaks down the requirements stated in the PLA into more fine-grained ones according to the business process and represents them as a hypergraph.

Security requirements are identified for each activity of BP_h and connected with a top QoP node (PLA). We show this as a hyperedge from the requirements for the activities to the top QoP node for "flow", "sequence", "switch" and "while" patterns. Then we repeat the process for each activity and its sub-process. If design alternatives for the decomposition exist they are represented as several hyperedges.

Different partners to whom some services (parts of the BP) are outsourced have various level of trust. This fact also impacts identification of target metric values. A contractor may trust one partner that the defined metrics for the activity will be achieved and not trust another one. We use the following strategy to take this fact into consideration: if the contractor does not trust a partner that some QoP requirements will be achieved he should increase the estimated bound of the external metrics. Now the contractor may trust more the partner since the requirements is more likely to be met. In the hypergraph a partner is represented as an extra node between the target node and source ones or simply as a node connected with the target node if the subprocess for the outsourced activity is not known. If there are several partners who fulfill the same activity we use one hyperedge, when several alternative partners are connected to the target node with several distinct hyperedges. The algorithm for the process is shown in Figure 1. It takes a set of business processes S_{BP} and a set of activities A and returns a QoP hypergraph $H = (N, E)$ where N is a set of nodes and E is a set of hyperedges.

EXAMPLE 1. *Let us consider the following e-banking scenario. A holding company (customer) outsources task of providing a loan to one of its subsidiaries (contractor). The procedure is implemented using Web services. The subsidiary specifies a business process shown on the left side in Figure 2. The contract between the partners states that no more than 10 frauds may occur per one year of providing the service. To determine if it can meet this requirement the subsidiary first creates a QoP hypergraph as it is shown in Figure 2. The defined process is not finite because there are several design alternatives. First, the subsidiary has to select the credit bureau it will invoke to receive trustworthiness rating of a client. Second, the subsidiary may prepare a loan for all clients in the same way, or to prepare a loan for ordinary clients when the procedure for VIP persons is provided by a special department. Note, that the alternatives are shown in the figure as separate hyperedges leading to the same target node. The process of VIP department is known because it is under the subsidiary's control while credit bureaus are black boxes for the subsidiary.*

3.2 Phase 2. Propagation function assignment

Now we define semantics for QoP hypergraph. For each hyperedge a weight that shows contribution of a source node to the target one is assigned. Weights of edges connecting partners with a target node specify the level of trust between the delegator and the delegatee. Since in our case each source QoP node contributes differently to the target QoP one we use intermediate nodes between source and target nodes. The weights are assigned to the edges which connect source and intermediate nodes and the weights for the edge between the latter and target nodes are neutral (e.g., 1). We do not depict the nodes in the figures to avoid unnecessary complexity.

For all nodes we assign a tuple $\langle M_{QoP}, f_{QoP} \rangle$ where M_{QoP} is a vector of metric values which *can* be achieved if a specific QoSS is applied; f_{QoP} is a propagation function which computes a set of metric values M_{QoP} of the target node taking source nodes' M_{QoP}s and corresponding weights as arguments: $f_{QoP} : 2^W \times 2^{M_{QoP}} \mapsto M_{QoP}$. This function is different for the four basic structural patterns but it is defined in the same way for the same pattern. The functions depend on type of requirements and we are going to specify them in the future work. If an activity is outsourced the meaning of the function is how security requirements are changed according to trust level of the partner. These functions are determined by security staff using their experience, events history and modern trends.

3.3 Phase 3. Security services identification and decomposition

In this phase a contractor identifies security services which he has to provide to achieve requirements stated in the PLA. First of all, security services which can be implemented or which are already in place are determined. For each security service a set of security service parameters ($QoSS$) is

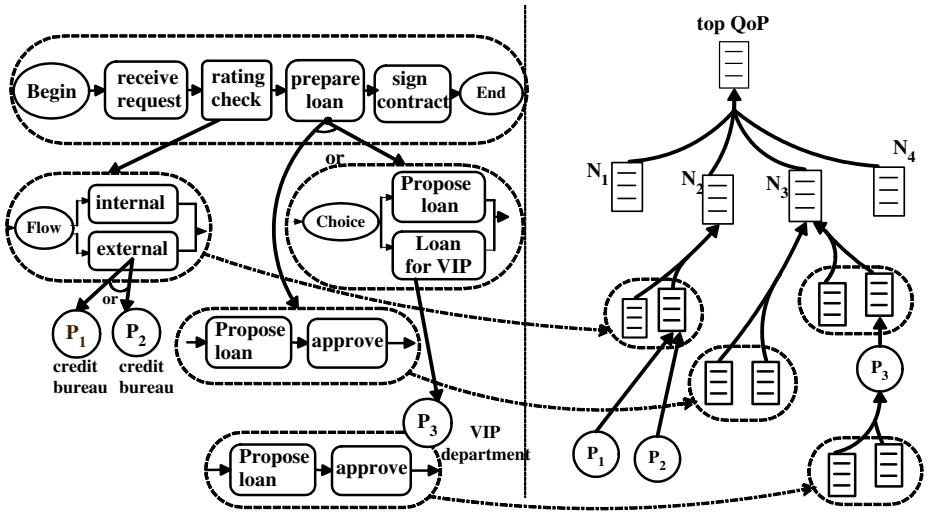

Figure 2: Building QoP hypergraph

Build_QoP_Hypergraph
input S_{BP}, A
  Add a new node QoP to N;
  $New\_Branch(S_{BP}, A, S_{BP}[1], QoP)$;
    `//Start with` BP_h $(S_{BP}[1])$
output N, E

New_Branch
input $S_{BP}, A, BP, TargetQoP$
  for all activities $A[j]$ **in** BP
    Add a new node QoP to N;
    Add node QoP to $SourceQoP$;
      `//set` $SourceQoP$ `is a tail of an edge`
    if the activity $A[j]$ is delegated **then**
      for all alternative sets of partners P_{alt} for $A[j]$
        `//for all edges connecting a set of`
        `//partners` P_{alt} `and target activity` $A[j]$
        for all partners p from set P_{alt}
          Add a new node QoP_1 to N
          Add node QoP_1 to $SourcePartner$
        `//set` $SourcePartners$ `is a tail of an edge`
        `//connecting a set of partners and` $A[j]$
          for all alternative BPs $S_{BP}[k]$ of p for $A[j]$
        `//`p `may fulfill` $A[j]$ `in different ways`
            $New\_Branch(S_{BP}, A, S_{BP}[k], QoP_1)$
        end
        Add an edge from $SourcePartner$ to QoP_1 in E
      end
    else
      for all refining BPs $S_{BP}[k]$ for activity $A[j]$)
        $New\_Branch(S_{BP}, A, S_{BP}[k], QoP)$
      end
    end
  Add an edge from $SourceQoP$ to $TargetQoP$ in E
  end
output N, E

Figure 1: QoP hypergraph building algorithm

determined. These parameters are internal security metrics of the service. Each compound service is decomposed in a similar way as it is shown in the first phase, so at the end we have a set of disjoint QoSS hypergraphs. A propagation function is assigned to each QoSS node which denotes how source security services contribute to the target one.

The contractor links potential security services with leaf QoP nodes which can be achieved if the countermeasures are installed (Figure 3). These links show if the countermeasures help to satisfy a requirement ("+" mark) or deny it ("-" mark). For leaf QoP nodes we assign a propagation functions similar to the one for other QoP nodes. For those

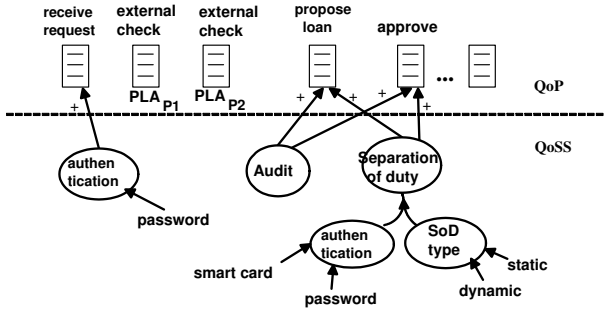

Figure 3: QoSS contribution.

leaf nodes which are delegated to other partners metric values can be taken from the corresponding PLAs. In case all tasks are outsourced (the contractor is a Web Services orchestrator) the methodology will choose those partners with which the overall process has the best protection level.

EXAMPLE 2. *In our example the security staff of the subsidiary have defined the the following security controls to reduce number of frauds: authentication of the client, audit of employees activity and separation of duty (to avoid approval*

of the loan by the same person that proposes it). Note, that for "external rating check" activity metric values are taken from PLAs of the credit bureaus.

3.4 Phase 4. A reasoning algorithm.

We apply a reasoning algorithm for testing different security configurations and determination of the best one. The contractor chooses a set of security services he is going to provide and determines security parameters of the leaf QoSS nodes. Using QoSS propagation function top security services are derived. Then the metrics for each leaf QoP node are calculated or determined according to PLAs for outsourced services. Now we have a classical problem of finding the shortest hyperpath in a hypergraph for which efficient algorithms have been proposed (e.g., [3]). Note, that these algorithms can be used only for those metrics for which QoP propagation functions are superior/inferior (e.g. number of attacks per execution). In the future work we are going to adopt the algorithms for other metrics (e.g., number of attacks per month). Finally, we receive the best value of the top QoP node. If the calculated protection level is less than the one agreed in the PLA with a client then another security configuration is tested. The process may be automated (to avoid manual correction of security parameters) but this direction requires further investigation such as definition of satisfaction function and security parameter correction mechanism.

4. RELATED WORK

There are a few papers which tackle the issue of security requirements in business outsourcing. One of the first papers discussing security SLA in a large enterprise is [9]. The main idea is to check compliance the system with fifteen security domains split into best practices. For each best practice the security service level is determined and added to the SLA (yet it does not consider outsourcing). Casola et. al. [5] extend the security decomposition to compare two SLAs or to find a security SLA which is the closest to the desired one. A similar idea was applied to evaluation of Web Service security by Wang and Ray [14]. Karjoth et. al. [11] claimed that security requirements must be reflected in the contract. *Trusted Virtual Domains (TVDs)* [8] are intended to connect a number of remote trustable virtual processing environments in one secure network. Security operational policy (accord of PLA/SLA), which is obligatory for every environment, are used. This technology can be applied to client-contractor interaction when one side (most likely, a contractor) allows another one to use its TVD.

5. CONCLUSION AND FUTURE WORK

In this work we have described the methodology which helps a contractor to determine the security system configuration that fulfills the requirements negotiated with a client. The methodology binds internal security requirements useful for a contractor with the external ones understandable by a client. It also allows a contractor easily recalculate security level if changes in a system configuration occur.

In future work we are going to define a propagation function for three basic business process constructs. We are also going to implement the algorithm adopted for chosen functions and test effectiveness and correctness of our approach.

6. ACKNOWLEDGEMENTS

I would like to thank my advisor Fabio Massacci for invaluable help in conducting this research.

7. REFERENCES

[1] A. Andrieux et. al. *Web Services Agreement Specification (WS-Agreement)*. Global Grid Forum, 2 edition, August 2004.

[2] B. Atkinson et. al. *Web Services Security*. Microsoft, IBM, VeriSign, 1.0 edition, April 2002.

[3] G. Ausiello, G. F. Italiano, and U. Nanni. Optimal traversal of directed hypergraphs. Technical Report TR-92-073, Pisa University and Monreal University, Berkeley, CA, 1992.

[4] S. A. Butler. Security attribute evaluation method. Technical Report CMU-CS-03-132, Carnegie Mellon University, May 2003.

[5] V. Casola, A. Mazzeo, N. Mazzocca, and M. Rak. A SLA evaluation methodology in Service Oriented Architectures. In *Proceedings of the 1st Workshop on Quality of Protection.*, Milan, Italy, 2005. Springer-Verlag.

[6] G. Della-Libera et. al. *Web Services Security Policy Language*. IBM and Microsoft and RSA Security and VeriSign, 2005.

[7] J. Eloff and M. Eloff. Information Security Management - A New Paradigm. In *Proceedings of the South African Institute of Computer Scientists and Information Technologists*, pages 130 – 136, 2003.

[8] J. L. Griffin, T. Jaeger, R. Perez, R. Sailer, L. van Doorn, and R. Cáceres. Trusted virtual domains: Toward secure distributed services. In *Proceedings of the 1st Workshop on Hot Topics in System Dependability*, Yokohama, Japan, June 2005.

[9] R. Henning. Security service level agreements: quantifiable security for the enterprise? In *Proceedings of the 1999 Workshop on New security paradigms*, pages 54–60. ACM Press, 2000.

[10] Y. Karabulut, F. Kerschbaum, P. Robinson, F. Massacci, and A. Yautsiukhin. Security and trust in it business outsourcing: a manifesto. In *Proceedings of the 2nd International Workshop on Security and Trust Management. To appear.* Electronic Notes in Theoretical Computer Science, 2006.

[11] G. Karjoth et. al. Service-oriented assurance comprehensive security by explicit assurances. In *Proceedings of the 1st Workshop on Quality of Protection.*, Milan, Italy, September 2005. Springer-Verlag.

[12] D. D. Lamanna, J. Skene, and W. Emmerich. SLAng: A Language for Defining Service Level Agreements. In *Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*, pages 100–18. IEEE Computer Society Press, 2003.

[13] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, 1999.

[14] Y. Wang and P. K. Ray. Evaluation methodology for the security of e-finance systems. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service* . IEEE Press, 2005.