THREAT RISK MODELING FOR WEB SERVICES

AVASOF

ABSTRACT

Web service - Sharing the business logics, data through a programmatic interface across the network (internet/intranet). It is a reusable application component and ideally anyone can use it in their application. Web service is one kind of distributed computing and these are self-described and self-contained components. Due to its features like flexibility and interoperability anyone can access web service that gets published in internet. High degree of flexibility offers low degree reliability and testability in the aspect of quality attributes for functional verification. In non-functional aspect, risk of insecurity in web services is considered as a key highlight because of the basic fact that web services are made open in the internet network, no matter security standards defined. Thus Security testing plays an essential role in web services that includes Authentication and Authorization of users prior to service access, Integrity & Confidentiality of requested data between the services etc., Threat Modeling is an engineering technique to identify the security objectives, relevant threats and vulnerabilities so as to perform countermeasures. Threat model helps prioritizing risk mitigation and drive the security testing effort. Though threat modeling for a web application that uses web service is available in current IT filed, threat modeling for an independent Web services (i.e. at unit level) is an opportunity area to work upon. This paper focuses on deriving threat risk modeling specific to a web service at unit level and leveraging it to its deployments patterns. The derived strategy for threat risk modeling will be unique for the standard web service model.

KEYWORDS - Threat modeling, Web service, Risk rating
# Table of Contents

1. Introduction.................................................................................................................. 3

2. Basics of Web service........................................................................................................ 3

3. Threat Risk Modeling Process.................................................................................. 4
   3.1 Web service Architecture and Quality attributes ................................................. 5
   3.2 Web service Integration with Application .............................................................. 6
   3.3 Identify Security Objectives .................................................................................... 7
   3.4 Identify threats in line with security objectives ...................................................... 8
   3.5 Threat mapping ........................................................................................................ 8
   3.6 Risk Rating ............................................................................................................. 10

4. Conclusion .................................................................................................................. 11

5. References................................................................................................................. 11
1. INTRODUCTION
As technology advances to meet the demanding need for addressing technical and management gaps in IT, issues on insecurity also evolves in direct proportional relationship. Key challenges in this aspect for the testers is to first learn the evolved technology and then to assess the system on either functional or non-functional aspect. In non-functional aspect, threat modeling is a necessary step to perform security assessment for the application under test and the same applies for application that uses web service. The invoking pattern of web service in terms of usage perspective by the calling application can have ‘N’ possibilities. So instead of performing threat modeling for all combinations of calling applications, it is recommended to perform threat modeling at web service level and then to leverage the modeling for the calling application pattern as well. In this context, a threat is defined as something capable of exploiting vulnerability that lead to actual loss specific to an asset defined for web service. Threat risk modeling of web service will identify potential threats and outline the degree of risk possessed by each of the identified threats. In order to accomplish the above process, one need to understand the basics of web service and its high level architecture with different combination patterns a web service will be accessed by the calling application. Thus threat risk modeling of web service is performed by identifying degree of risk possessed by the imparting threats against the identified assets specific to web service.

2. BASICS OF WEB SERVICE
Web Services are software components that allow the transmission of data across internet using pervasive, standard-based web technologies. In simple words a business solution or an application published in internet by an organization are made available to any other organization whenever required; provided the service has been shared by the organization owning it. Thus web service is essentially an XML-messaging based interface to some computing resource. The web services protocol stack consists of:

- Some transport layer protocol, typically HTTP
- An XML-based messaging layer protocol, typically SOAP
- A service description layer protocol, typically WSDL
- A service discovery layer protocol, typically UDDI
To make the service available to the Service Consumer, the Service Provider should publish the web service in the UDDI (Service Directory). The Service Consumer who needs a particular service sends a query to the UDDI which in turn responds to the Service Consumer with a WSDL document of the relevant web services. Depending upon the information provided in the WSDL, further interaction takes place between the Service Consumer and the Service Provider by means of SOAP messages.

3. THREAT RISK MODELING PROCESS

Threat Risk Modeling is an incremental process that starts during the early phases of application development and it should be repeated as the application evolves. The process for threat risk modeling of Web services comprises of six main steps and these are detailed below.

1. Understand Web Architecture: As a first step, one need to understand the architecture of the web services including the four meta-models and its governing quality attributes.

2. Deployment model with possible Integration: Document the integration of the web services interactions with the details on architecture including application server, web server, database server and other possible external integration systems.

3. Identify Security Objective: Identify the valuable assets in line with integration and also for web services attributes.

4. Identify Threats: Identify the threats relevant to the identified security assets in Step3 for Web service attributes (i.e. at unit level).

5. Mapping: The identified security threats will be mapped with the deployment model of web service integration.
6. **Risk Rating**: Rate and priorities threats according to probability of occurrence, severity and detection that has dependency with the relevant application integration

### WEB SERVICE ARCHITECTURE AND QUALITY ATTRIBUTES

The diagram below depicts web service interoperability architecture that defines global elements of global web services without imposing any restrictions on how web service must be combined. Web service architecture is articulated as models along with its concepts and relationships that forms the quality attributes for a web service. These quality attributes of a web service describes minimal characteristics that are common to all web service with the variation in number of characteristics as specific to a web service. Models mainly focus on the interactions between three roles: Service provider, Service discovery and Service requestor.

**Message Oriented Model (MOM)** focuses on messages, message structure, and message transport and policy associated with it. This is characterized by agent that sends and receives, the structure of messages in terms of message header and body and mechanisms used to deliver messages like message exchange pattern and correlation. Briefly, MOM ensures proper interaction for web service agents (requestor & provider agents) using message oriented communication model.

**Service Oriented Model (SOM)** focuses on the relationships between an agent and the services by interpreting messages as requests for actions and as responses to those requests. The interpretation of messages happens as
an incremental and layered approach using the attributes like service description, service task, resource policy and others as listed in the diagram.

**Resource Oriented Model (ROM)** focuses on the key features of resources such as the ownership of resources, resource description, policies associated with resources and its representation. A resource is anything that is important enough to be referenced a thing in itself. So a resource is something that can be stored on a computer, a document, stream of bits, record in database and result of a simple algorithm. In Web service, resources are important internally to the architecture as web service is best understood as a resource in the context of web service management and in terms of policy management. In external perspective resources are important metaphor for interpreting the interaction between requestor and provider entity.

**Policy Model (PM)** focuses on constraints on the behavior of agents and services and to represent security concerns, quality of service. Here the policy model set some constraints to the rest of three models. Domain is a group of agents and/or resources; it defines the scope of application of one or more policies.
3.2 WEB SERVICE INTEGRATION WITH APPLICATION

The above diagram represents possible interaction of web applications (that in turn can have different possibilities as depicted) with Web service that also can have different deployment models as captured in the diagram. The illustrations of these interactions keeping web application as front end are detailed as below points.

Web server functionality abstraction: Web service can be used to build rich client configurations whose configuration lies somewhere between a thin client (relies on network-distributed system) and fat client (relies on resources installed locally). These rich client forms a networked computer that has some resources installed locally but also depends on other resources in the distributed network. User can download client-side application components such as Java Applets from the web server. These downloaded components will then interact with the web server using web services.

Application server functionality abstraction: If Business workflows, components and entities related functionalities can be translated as Web service. For example functionality of a business component like ‘Make
Payment’ can be made as web service that can accessed by two different businesses intended applications like Bus booking and Cinema booking applications.

**Interaction with external systems:** Legacy application interaction functionality or any other external interactions like LDAP by a web application can be translated as a web service that in turn can be used by the application server.

**Multistage processing:** Web services support an asynchronous messaging model. A single request can traverse multiple intermediaries before reaching its final destination. For example, an authentication server as intermediary can authenticate the SOAP message before its arrival at the application server.

### 3.3 IDENTIFY SECURITY OBJECTIVES

There are many security challenges that exist while adopting web services. In a broader perspective, the objective is to create an environment where message level transactions and business processes can be conducted securely in an end-end fashion. There is also a need to ensure that messages are secured during transit with or without mediators. Like any other systems, web service needs protection at multiple levels based on the entry points identified. These expected protections become relevant security objectives of the web services as captured below.

**Protection for Independent Web service**

The security objective for a web service will be on the aspect of who are the authenticated and authorized users to access the web service or its internal functionalities, mechanism to ensure if the associated internal attributes like SOAP header, body, and policy of web service are not altered by external factors. Additional factor will be to ensure if discovery mechanism is secured that enforces policies to govern and discover a particular service.

**Protection for calling application**

The security objective of calling application that access web service will be on the aspect of data integrity, transactions integrity, end-end integrity, confidentiality of messages, non-repudiation together with a secured communication channel. One more added factor will be to ensure if trust mechanism is not altered or overlooked that guides the requestor agent to take a decision on whether or not to trust a particular service.

**Protection for Web service Interacting components**

The web service may need to interact with any external components so as to execute its intended business logic that in turn will return results to the calling applications. In this aspect the objective will be to ensure if the proper authentication & authorization is ensured to access external components and no changes are inserted to the interacting components internal logic/ architecture.
3.4 IDENTIFY THREATS INLINE WITH SECURITY OBJECTIVES

From the perspective of documented web service architecture and its integration along with the relevancy to the captured security objectives, high level threats details are given below

Threats for Independent Web service

Permission guard & Audit Guard Violation

Conflicting policy definitions with authentication/authorization definitions

- Message alteration
- Unauthenticated / Unauthorized access
- Man – in- the- middle attack
- Spoofing
- Denial of service attacks
- Replay attacks
- Message exchange pattern or orchestration definition violation
- Invalidated inputs from calling applications
- Buffer overflow in case web services incorporating calculation logic

Threats for calling application and interacting components

- Data integrity issues
- SQL injection from Web service return value
- Invalidated re-directs and forwards from invoked web service
- Insufficient transport layer mechanism

3.5 THREAT MAPPING

The identified threats are grouped for the models identified in web service architecture and its integration patterns as captured below

<table>
<thead>
<tr>
<th>Threat Focus 1</th>
<th>Message Oriented Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat Targets</td>
<td>Message Body Message</td>
</tr>
<tr>
<td></td>
<td>Header</td>
</tr>
<tr>
<td></td>
<td>Message Transport</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Possible Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Insecure Direct Object references Insecure</td>
</tr>
<tr>
<td>2. Cryptographic Storage</td>
</tr>
<tr>
<td>3. Buffer overflow</td>
</tr>
<tr>
<td>4. XML Injection</td>
</tr>
<tr>
<td>5. Insufficient Transport Layer Protection</td>
</tr>
<tr>
<td>6. Replay Messages</td>
</tr>
</tbody>
</table>
### Nature of Attack and resultant outcome (in the order of captured threats)

1. A compromised intermediary may modify the message attributes of a Web service to affect the data integrity of a system.
2. Fail to protect the sensitive data in SOAP message that may alter message exchange pattern
3. A large value is placed in parameter inside a SOAP body and sent to a receiving application that may not be prepared to handle an overly long string
4. The attacker may try to inject flaws in XML messages like deleting or modifying a part of the message, inserting extra information in the messages, manipulating attachments, if any
5. Protection of web services data in SOAP messages from unauthorized disclosure and modification when it is transmitted
6. Attackers can intercept a message and will replay it to the targeted agent when message model lags time stamps and number sequencing of messages

### Threat Focus 2: Service Oriented Model

**Threat Targets**

- Service Description Service
- Role

**Possible Threats**

1. Insecure Cryptographic Storage
2. Insufficient Transport Layer Protection
3. Insecure Direct Object references
4. Improper/ Misleading Authentication

**Nature of Attack and resultant outcome (in the order of captured threats)**

1. Attack code can be sent via a SOAP message in a service to be later executed in a receiving application which can alter message exchange patterns, semantics or choreography of the described service
2. Protection of the service properties from an unauthorized disclosure and modification when it is transmitted
3. An intermediary may modify the properties of service role of a Web service to affect the defined bindings
4. Authentication is needed in order to verify the identities of the requester and provider agents for providing and consuming services. The threat will happen in case of intended changes by attackers or improper authentication predefined to a machine instead at a user level or authentication made for an entity instead for a message etc.,

### Threat Focus 3: Resource Oriented Model

**Threat Targets**

- Resource Identifier Resource
- Description

**Possible Threats**

1. Insecure Direct Object references
2. Insufficient Transport Layer Protection
3. Cross-site scripting

**Nature of Attack and resultant outcome (in the order of captured threats)**

1. An intermediary may modify the resource identifier of a web service as WSDL file that is sent to the client when tampered can mislead the client
2. Unauthorized disclosure and modification of the resource description when it is transmitted
3. Changes in the destination address or other processing while resending the processed data by compromising SOAP intermediary to intercept messages between web service requestor and ultimate receiver

<table>
<thead>
<tr>
<th>Threat Focus 4</th>
<th>Policy</th>
<th>Model</th>
</tr>
</thead>
</table>
| Threat Targets | - Domain  
- Audit Guard  
- Permission Guard | |
| Possible Threats | - XML Injection  
- Anonymous access  
- Improper Authorization | |
| Nature of Attack and resultant outcome (in the order of captured threats) | - The attacker’s hostile data can trick the policy description by the person or organization that manipulates the message, service or resource policies or manipulating the defined domains  
- As it is possible to implement service without identity that will lead to anonymous service access without revealing requestor’s entity. This will impose constraint to audit guard functionality of a web service  
- Authorization mechanisms control and allow permission to resources to access appropriate service and this can be altered in permission guard | |

<table>
<thead>
<tr>
<th>Threat Focus 5</th>
<th>Integration with other applications/systems</th>
</tr>
</thead>
</table>
| Threat Targets | - Web Application Legacy  
- Systems  
- Database Server  
- Mail Server  
- LDAP and other possible integration | |
| Possible Threats | - SQL/ LDAP/ XML Injection  
- Insecure Cryptographic Storage  
- Invalidated redirects and forwards  
- Elevation of privilege  
- Repudiation  
- Improper Delegation  
- Improper exception management  
- Cross-site scripting  
- Denial of Service |
<table>
<thead>
<tr>
<th>Nature of Attack and resultant outcome (in the order of captured threats)</th>
<th>1. Either Attack code can be sent via a SOAP message to be later executed in a receiving application or Web service not handling the injection</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Discovery of encryption keys or break the encryption data, which used to encrypt sensitive data (including client credit card numbers) in the a database server/ Web Application</td>
<td></td>
</tr>
<tr>
<td>3. Attackers can break the flow of processing of transactions or data flows that will lead to the violation of defined message integrity, transactions integrity or endend logical processing integrity</td>
<td></td>
</tr>
<tr>
<td>4. The attacker run the code on the victim’s machine and borrow the privileges (like user id and password) in a legacy systems / web applications by inducing buffer overflow and cross site scripting in the victim machine</td>
<td></td>
</tr>
<tr>
<td>5. Repudiation attacks permits malicious manipulation or forging the identification of new actions. This attack can be used to change the authoring information of actions executed by a malicious user in order to log wrong data to log files. If this attack takes place, the data stored on log files can be considered invalid or misleading.</td>
<td></td>
</tr>
<tr>
<td>6. In case of service roles relying on other services or interacting components, there will be inadequate delegation of permission to the interacting components or an attackers changing the permission settings</td>
<td></td>
</tr>
<tr>
<td>7. Any improper input validation or other data processing if not handled between the interacting components or return value of web service to its relevant interactions will lead to unhandled exceptions</td>
<td></td>
</tr>
<tr>
<td>8. Changes in the destination address or other processing while resending the processed data by compromising SOAP intermediary to intercept messages between web service requestor and ultimate receiver</td>
<td></td>
</tr>
<tr>
<td>9. Send a message with many encrypted elements or signed elements from more than one machine and prevent legitimate users of the service from the ability to use the service</td>
<td></td>
</tr>
</tbody>
</table>

Risk rating is performed by analyzing the documented threats. The actual risk rating for the threats is considered out of scope for this paper, since the influencing factors for risk rating has dependencies with the actual web service implementation and its integration functionality knowledge. Sample risk rating is derived below for some of the identified threats that are assumed to serve as guideline for other threats during real time usage. For risk rating one need to identify threat source and its relevant threats to rate the risks in terms of below attributes.

**Risk Likelihood** – It defines the probability of occurrence of imposed threat and can be rated with a value ranges as 1, 3, 7, and 9. Higher rating mean of imposed threat is most likely to occur and will provide an insight that specific threat can impose high risk to turn as loss

**Risk Severity** – It defines the impact of threat and can be rated with a value ranges as 1, 3, 7, and 9. Higher rating mean that severity of imposing threat is high and will provide an insight that specific threat can impose high risk to turn as loss
**Risk Detection** – It defines the detection capability of imposed threats and can be rated with a value ranges as 1, 3, 7, and 9. Higher rating mean that imposed threats are unlikely to detect with direct mechanisms and will provide an insight that specific threat can impose high risk to turn as loss.

**Risk rating** – This is the factor that multiplies the value ranges for risk likelihood, severity and detection as defined for the specific threat to access overall risk imposed by the threat.

**Sample Risk rating**

Sample risk rating is performed for a web application that interacts with web service. In this case the requesting service is to get some data from external databases through web service that in turn will update in the intended application. The requested data is only some general information about different products and nothing on financial/ personal record details. In this context, risk rating is derived as below.

<table>
<thead>
<tr>
<th>Threat Source</th>
<th>Threat</th>
<th>Risk Likelihood (L)</th>
<th>Risk Severity (S)</th>
<th>Risk Detection (D)</th>
<th>Risk Rating (L<em>S</em>D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat Focus 5</td>
<td>SQL injection</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>245</td>
</tr>
<tr>
<td></td>
<td>Invalidated re-directs</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Insecure cryptographic storage</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>27</td>
</tr>
</tbody>
</table>

### 4. CONCLUSION

Threat risk modeling is described for web service and its integration models. The web service architecture is detailed to explore on the possible threats. The main focus areas for threat mapping are identified as independent web service, Web application, web service integration components. Finally guideline to derive risk rating for the mapped threats is captured for a sample web service assessment.

### 5. REFERENCES

[1] Lieven Desmet, Bart Jacobs, Frank Piessens, and Wouter Joosen, “THREAT MODELING FOR WEB SERVICES BASED WEB APPLICATIONS”

[2] [http://www.w3.org/TR/ws-arch/#security_policies](http://www.w3.org/TR/ws-arch/#security_policies)