Abstract

The paper provides an overview of available web applications and Web Services security vulnerability models and proposes a classification of the potential Grid and Web Services attacks and vulnerabilities. This is further used to introduce a security model for interacting Grid and Web Services that illustrates how basic security services should interact to provide an attack-resilient multilayer protection in a typical service-oriented architecture. The analysis and the model can be used as a basis for developing countermeasures against known vulnerabilities and proposing security services design recommendations. The paper refers to the ongoing work on middleware and operational security in the framework of the European Grid infrastructure deployment project EGEE and related coordination groups.

1 Introduction

The area of Web Services and Grid security vulnerabilities and threats continues to be new for researchers and developers.

The paper presents an ongoing work that intends in its final result to provide recommendations to the security middleware developers how to address identified specific Web Services and Grid security vulnerabilities, in a first row, vulnerabilities of the basic security services that affect Grid applications security, i.e. authentication, authorisation, confidentiality and data protection, remote access and secure communication.

Presented here Grid security vulnerabilities and threats analysis is built upon existing security vulnerabilities models and classifications. It provides a basis for a proposed security model for interacting Web Services and Grids that can be used further in proper security services design and operational procedures development.

The paper outline: section 2 provides general overview of existing models and classifications for web application vulnerabilities and proposes a general classification of Web Services security threats/attacks. Section 3 extends this analysis to the specific attacks against interacting Grid and Web Services and proposes simple model that groups all attacks depending on their origin and target vulnerability. Section 4 proposes detailed model describing how security services are built into general service oriented architecture and how they interact to provide multilayer security for both Service/Resource and User/Requestor. Finally section 5 provides initial suggestions how identified threats and/or attacks can be addressed in security middleware design and operational procedures.

2 General approach and existing models for web application vulnerabilities

This section provides a short overview and a summary of existing approaches to web applications security vulnerabilities analysis and modeling.

The following Vulnerability-Incident life-cycle model provides illustration how vulnerability may become a potential security threat and further develop to an Incident:

Vulnerability => Exploit => Threat =>
Attack/Intrusion => Incident

Vulnerability is a flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy Exploit is a known way to take advantage of a specific software vulnerability Threat is a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm Attack is an assault on system security that derives from an intelligent threat Incident is a result of successful Attack

An attack is defined as an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the
security policy of a system. An attack may consist of one or more steps taken by an attacker to achieve an unauthorised result. A successful attack may lead to an intrusion and be further escalated as an incident [1].

From the life-cycle above we can understand how an attack is prepared and undertaken by attackers to target the application in general or with the specific vulnerability. The basic steps in attacker methodology are summarized below and illustrated in Fig. 1 [2]:

- Survey and assess
- Exploit and penetrate
- Escalate privileges
- Maintain access or Deny service
- Unauthorised use of Resource (including unauthorised access to information)
- Clean or forge track of activity

![Diagram of attacker methodology](image)

**Figure 1. Basic steps for attacking methodology.**

Emerging Web Services and Computer Grids open new kind of Security attacks and Incidents that can be defined as “white collar” attacks. Specifics of this kind of attack, from the point of view of applications and network protection, is that malifactor is interested in correct and smooth working of a target system or application, so that his or her activity doesn’t interrupt the normal system operation. Classically, white-collar crime or commercial crime involves crimes such as fraud, identity theft, etc. They are a lot easier to hide than other forms of crime and therefore it is much harder for the business to stop and the criminal justice system to deal with. The same may be applied to attack via interception and tampering Web Services communication. Incidents based on credentials theft will be even more difficult to discover at the earlier stage and track down to the originator. With high level of impersonation and use of the electronic identity and delegation in Grids and Web Services, the character of threats and security incidents will inevitably change with time.

There are few known projects and initiatives discussed below that mostly focus on vulnerability/threat analysis for web applications providing also recommendations for applications specific countermeasures but still leave attack/intrusion identification out of scope.

### 2.1 OWASP, WASC and EVDL vulnerabilities classification

The Open Web Application Security Project (OWASP) [3] proposed a commonly recognised classification of the web application vulnerabilities in 2003/2004. This proposal identifies the “Top Ten Most Critical Web Application Security Vulnerabilities”. The OWASP Top Ten represents a broad consensus about what are the most critical web application security flaws and vulnerabilities and actually establishes a minimum standard for web application security.

Another association of the security experts the “Web Application Security Consortium” (WASC) recently released the document on Web Applications threats classification that tends to establish common industry terminology and classification for known vulnerabilities [4].

A short summary of OWASP the most significant web application security vulnerabilities and their relations to major identified Web Services vulnerabilities can be found in the EGEE JRA3 documents [1, 2].

Proposed by OASIS Web Application Security TC the Enterprise Vulnerability Description Language (EVDL) is based on OWASP and WASC classifications and provides more detailed breakdown of the major identified vulnerabilities [5]. EVDL intends to become a comprehensive application security markup language whose primary goal is to facilitate communication about specific application security vulnerabilities, techniques for discovering those vulnerabilities, and measures to protect against those vulnerabilities.

### 2.2 Web Application Security Threats Model and Classification by Microsoft

The guide by Microsoft “Improving Web Application Security: Threats and Countermeasures Roadmap” [6], published in 2003, provides comprehensive analysis and recommendation how to build hack-resilient applications. A hack-resilient application is one that reduces the likelihood of a successful attack and mitigates the extent of damage if an attack occurs. A hack-resilient application resides on a secure host (server) in a secure network and is developed using secure design and development guidelines. The guide addresses web application security [1].

---

1 The person with malicious intents, e.g. intruder or attacker in the security incident
across the application tiers and at three basic layers: the network, the host, and the application. A weakness in any tier or layer makes the application vulnerable to attack.

Network security is provided by such network infrastructure components as Routers, Switches and Network Firewalls that provide communication security at Layers 1-4 - Physical, Data, Internet, Transport between communicating network nodes. Most of current applications are built for secure operation in such uncontrolled network environment as global Internet and rely on existing network security services and mechanisms to provide host-to-host network level security.

The host security concerns additional measures to protect normal/secure host operation. Host can be protected from the network by Network Firewall, which can isolate the host from undesirable network traffic and open Internet exposure but still cannot protect the host from attacks that target applications’ network services that normally are run at the host hosting application. More structured presentation of the host security components includes the following components:

- Protocols and Ports that provides network access and communication services for applications.
- Common OS Services
- Files and Directories
- User Accounts and privileges
- Registries
- Auditing and Logging
- Patches and Updates management

Application Vulnerabilities in the guide [6] are categorised similar to the OWASP Top Ten vulnerabilities and include: input validation, authentication, authorization, configuration management, sensitive data, session management, cryptography, parameter manipulation, exception management, auditing and logging.

2.3 Proposed Web Services threats/attacks classification

This section provides a short overview of the Web Services vulnerability analysis and classification proposed by authors in the EGEE MJRA3.4 document [1] and extends it with the potential WS-Security vulnerabilities. This classification also refers to the earlier works by Forum Systems [7] and Spire Security [8].

Web Services attacks can be classified in the following way depending on targeted vulnerability groups:

XWS1 - Web Service interface (WSDL) probing attacks. WSDL as an advertising mechanism for Web Services describes the methods and parameters used to access a specific Web Service, and in this way exposes the Web Service to possible attacks by providing a potential attacker initial information about how to access a specific web service. Additionally, improper exceptions/errors handling by service hosting platform may provide important information about the internal service structure that can be used by attacker.

XWS2 - Brute force XML parsing system attacks. XML parsing is a resource and time consuming process. Many real world applications may allow complex or voluminous XML input which may overload the XML parsing system resulting in Denial of Service (DoS).

XWS3 - Malicious Content attacks. XML documents may contain malicious parsing or processing instructions (XML Schema extensions, XPath or XQuery instructions, XSLT instructions, etc) that may alter the XML parsing process, or malicious content that may carry threats to back-end applications or hosting environments (application specific commands with the malicious code addressing known vulnerabilities in applications, e.g. buffer overflow, Unicode based vulnerabilities, etc.).

XWS4 - External Reference attacks. This group of attacks is based on the generic ability of XML to include references to external documents or data types. This group of attacks is distinguished from malicious content attacks by involving external resources or sites that can be manipulated by an attacker. Poor configuration, or improper use of external resources can be exploited by hackers to create DoS scenarios or information theft.

XWS5 - SOAP/XML Protocol attacks. The SOAP messaging infrastructure operates on top of network transport protocols, uses similar services for delivering and routing SOAP messages, and therefore can be susceptible to typical network/infrastructure based attacks like Denial of Service (DoS), replay or man-in-the-middle attacks.

XWS6 - XML Security Credentials tampering. XML credentials and assertions are used for requestor and service authentication, authorisation and session or state management. They can be in the form of XML wrapped user certificates, signed and/or encrypted XML documents, or session key established during secure context negotiation. Suggested vulnerabilities and threats include XML Signature and secure XML content manipulation, XML credentials replay, application session hijacking.

XWS7 - Secure key/session negotiation tampering. This group of attacks can be a result of poor WS-Security implementation, poor key generation or key management, weak or custom encryption. Unicode content manipulation may be used for credentials forging [9].

For completeness, we should also mention underlying transport protocol attacks. These attacks are not related to
Web Services but directly affect the reliability of SOAP based communications; they should be addressed at the network infrastructure level by implementing TLS, IPSec, Firewall, etc.

Table 1 summarises discussed above Web Services vulnerabilities and provides their mapping to OWASP classification [1].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>XWS1 – Web Services Interface probing</td>
<td>WSDL scanning, WSDL parameters tampering, WSDL error interface probing</td>
<td>A1 - Unvalidated Input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A2 – Broken Access Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A3 – Broken Authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A7 - Improper Error Handling</td>
</tr>
<tr>
<td>XWS2 – XML parsing system</td>
<td>Recursive XML document content, oversized XML document</td>
<td>A1 - Unvalidated Input</td>
</tr>
<tr>
<td>XWS3 – Malicious XML content</td>
<td>Malicious code exploiting known vulnerabilities in back-end applications, viruses or Trojan horse programs, malicious XPath or XQuery built-in operations, malicious Unicode content</td>
<td>A1 - Unvalidated Input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A5 - Buffer Overflows</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A6 - Injection Flaws</td>
</tr>
<tr>
<td>XWS4 – External reference attacks</td>
<td>Malicious XML Schema extensions, namespace resolution manipulation, external entity attacks</td>
<td>A1 - Un validated Input</td>
</tr>
<tr>
<td>XWS6 – XML security credentials tampering</td>
<td>XML Signature manipulation, secured XML content manipulation, Unicode content manipulation, XML credentials replay, application session hijacking</td>
<td>A3 – Broken Authentication and Session Management</td>
</tr>
<tr>
<td>XWS7 – Secure key/session negotiation tampering</td>
<td>Poor WS-Security implementation, poor key generation, poor key/trust management; weak or custom encryption</td>
<td>A2 – Broken Access Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A3 – Broken Authentication and Session Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A10 - Insecure Configuration Management</td>
</tr>
</tbody>
</table>

Note: *) There is no direct mapping between two classifications because OWASP is focused on application side vulnerabilities and attacks and proposed Web Services attacks/threats classification describes interacting Web Services.

3 Security Attacks Groups in Interacting Grid and Web Services

Proposed in this section security threats model intends to address known vulnerabilities and concerns in current Grid middleware implementation and provide a general approach to both security services design and operational security. It is based on ongoing work in major European Grid related projects, in particular, EGEE and GridPP [2, 10].

This work can be considered as a first step to further creation of more detailed security models for basic middleware security services, in particular:
- Authentication system
- Authorisation system
- Credentials use and management
- Remote access and communication
- Data protection

Fig. 2 below provides a general model of interacting Grid and Web Services, represented by the Requestor/User and Service/Resource, and identifies the following threat/attack groups:

**UCA - User Credentials Attacks** comprise of possible attacks originated from and based on user credentials theft or compromise that may happen as a result of user system compromise or by intercepting user-service communication, if user credentials are not protected enough. User impersonation may happen without direct compromise of user credentials but with more complicated playing with the processes of user AuthN and AuthZ to the remote service, if AuthN and AuthZ sequences are not protected enough in respect of message and credentials integrity and confidentiality, and proper secure context management.

**WIA - “Wire” Intelligence Attacks** include a wide spectrum of attacks that can happen if service-level communication is not protected enough against eavesdropping and interception. Beside basic service request and response, Web Services communication
includes service discovery, AuthN/Z stages, security context negotiation and exchange, including session management. Most threats in WIA group come from potentially uncontrolled environment messages may pass, especially if end-to-end service communication involves SOAP messages routing and intermediate processing. Communication and messages compromise and manipulation may lead to such classes of attacks as “Man in the middle” (MITM), credentials compromise and/or replay, session hijack, SOAP routing detour, and as well as attributes/credentials probing and brute force attacks.

**MIA - Malifactor Initiated Attacks.** This group of attacks can be undertaken by a potential attacker using both traditional and Web Services specific techniques that include WSDL probing, malicious XML content, brute force and dictionary attacks to bay-pass site security services, and traditional Denial of Service (DoS) attacks that may target all components of the site services stack. It is even more difficult to avoid this type of attacks against Web Services because traditional network and host protection tools, like Firewalls, are transparent to SOAP communications.

**SMA - Site Management Attacks** include possible attacks that can be caused by improper site security services configuration and management: insufficient AuthN and AuthZ credentials verification including security context verification, improper key and privileges management and control, improper error handling that may disclose internal information about service operation, and also insufficient or insecure logging that may allow an attacker to hide or forge its activity.

**ESA - End Service Attacks** target known vulnerabilities in the end-service. They use different techniques to construct malicious input content, e.g. XML/SQL injection, external references in XML schema and XML documents, internal and external cross-references with XPath and XSLT instructions. Attacker may intend to violate suggested quota or acceptable use of the resource what may be prevented by proper access control and accounting. End service application can be a target and a mediator of viruses and worms carried over some types of unchecked input, and therefore antivirus protection should also be considered for Web Services applications.

---

As mentioned before, Web Services and Grid are also susceptible to all underlying network and hosting
environment attacks, which protection is reasonably well covered with available products. Table 2 provides more detailed breakdown of the identified attacks and vulnerability groups together with their mapping to the proposed earlier Web Services attacks classification.

### Table 2. Threats/Attacks groups in interacting Web Services and Grids

<table>
<thead>
<tr>
<th>Threats/Attacks groups</th>
<th>Threats/Vulnerabilities</th>
<th>XWS threats mapping</th>
</tr>
</thead>
</table>
| UCA – User Credentials Attacks | • Credentials theft  
• Credentials compromise  
• User impersonation | XWS6 – XML credentials tampering  
XWS7 – Secure key/session negotiation tampering |
| WIA – “Wire” Intelligence Attacks | • Network eavesdropping  
• “Man in the middle” (MITM)  
• Brute force  
• Credentials compromise  
• Replay/Session hijack  
• Replay attributes probing  
• XML/SOAP protocol | XWS5 – XML Protocol attacks  
XWS6 – XML credentials tampering  
XWS7 – Secure key/session negotiation tampering |
| MIA – Malifactor Initiated Attacks | • WSDL probing  
• Malicious content  
• DoS  
• Brute force  
• Dictionary attacks | XWS1 – Web Services Interface probing  
XWS2 – XML parsing system  
XWS3 – Malicious XML content |
| SIA – Site Management Attacks | • Configuration vulnerabilities  
• Insufficient authentication and authorisation credentials verification  
• Improper key/trust management  
• Improper privilege management  
• Improper password/ credentials recovery  
• Improper error handling  
• Insecure audit/log | XWS7 – Secure key/session negotiation tampering  
XWS6 – XML credentials tampering |
| ESA – End Service Attacks | • Malicious input  
• XML/SQL injection  
• Dynamic XML  
• Resource misuse and quota violation  
• Viruses and worms | XWS4 – External reference attacks  
XWS3 – Malicious XML content |

4 Security model for interacting Web Services and Grid

4.1 Resource/Service Security Zones and Access Control Management

In the Grid services architecture (GSA) (as well as in the general Service Oriented Architecture (SOA)), the middleware provides a media for conveying a service request and delivering a service (or its product) in a controlled and secure way to the requestor. In such a model, the service or resource is placed at the back-end of interacting components and sub-systems. Middleware provides the hosting environment and required security services that ensure that service is delivered to the authorised user/entity and in the controlled secure way.

To address identified above vulnerabilities and have an instrument to analyse security vulnerabilities and develop necessary countermeasures against possible attacks, there was a need to create a new security model that represent interacting Grid and Web Services and address security issues at multiple application layers/tiers.

Figure 3 illustrates how major access control components interact in a typical GSA/SOA to provide multilayer security protection. It is based on typical implementation using container or application server for hosting Web Services based applications and provides a structured view of the Resource site security services. The following security zones are defined for the Resource/Service site:

**Zone R0** – zone controlled by the Resource itself that also includes local data storage and local file system; this is
the zone of the Resource trust level.

**Zone R1** – zone that includes Resource agent or interface and other sub-systems controlled and trusted by the Resource and can work under administrative privileges; this also includes the policy that is specified by the Resource and stored in the Policy Authority (PA). The Resource agent can also use its own access control service that is not exposed in the SOA relations/description.

**Zone RA and Zone RAA** – zones protected respectfully by Requestor and request authentication and authorisation. PDP (Policy Decision Point) as a central policy based decision making point, PEP (Policy Enforcement Point) providing Resource specific authorisation request/response handling and policy defined obligations execution, PAP (Policy Authority Point) as a policy storage (in general, distributed), and a AA (Attribute Authority) that manages user attributes and, in particular, for Grid applications can be VO management service (VOMS) [11].

**Zone RN** – zone that includes network access facility and actually open to the world; it may also contain the Firewall that is controlled by the Firewall policy and protects the Resource site from the external attacks against the network components and malicious input to the Resource services.

It is important to note that the Requestor or request authentication can be done as a separate procedure before authorisation or as an initial step/stage of the Requestor/Subject verification during authorisation. In the distributed access control infrastructure in order to optimise performance the Authorisation service may also issue authorisation tickets (AuthZTicket) that confirms access rights and is based on positive decision of the Authorisation system and can be used for granting access to the following similar requests that match AuthZTicket. However, to be consistent, AuthZTicket must preserve full context of the authorisation decision including AuthN context/mergent and policy reference.

Figure 3. Service/Resource site security zones.

Proposed security zones definition can be applied to both distributed and local zone related security services such as Authorisation or Policy enforcement and Authentication however their relation to the specific security zone should be maintained by proper trust relations or credentials path.

Depending on particular implementation. AuthN and AuthZ services can be implemented as part of application server or servlet container, e.g. in a form of message level filters, SOAP interceptors, etc., or run as an application component or separate services in the container.

Proposed security zone model extends other existing
models, such as the URL Security Zones used in Microsoft Internet Explorer security model [12] or security realms concept used in the Java Servlet specification [13] and implemented in the popular servlet container Apache Jakarta Tomcat [14], and provide better granularity for consistent security analysis of XML Web Services and Grid applications.

4.2 Requestor/User Security Zones and Credentials Management

Consistent security in interacting WSA/Grid services depends on proper requestor/user credentials management. Requestor/user in their interaction with the Grid/Web Service can be represented by their browser or other type of client, which however will require a common type of container that can be a browser or a servlet container like Tomcat. The client can act as a Requestor/user proxy in accessing remote service and needs to handle both user own credentials and temporal credentials provided by the service as a confirmation of user submission to the service.

For the formalisation purposes, we can specify a Requestor/user security zone model similar to the resource one:

- Zone A – Internet zone open to open Internet
- Zone B – browser or container cache for cookie, applets and session ID/data
- Zone C – user client/proxy storage that can store temporal application data or temporal user credentials, in particular, proxy-certificate used in Grid applications
- Zone D – local credentials storage that is protected by local file system and require special application to be accessed by user agent or Web Services application; normally, local user credentials can be protected by password
- Zone X – includes external credentials storage which also requires special tool to be accessed by the user or application

User can use different types of credentials that have different level of protection. However, to be submitted to the service or resource s/he must posses the credentials that prove their identity and, additionally, assigned attributes in the form of groups, roles or other privileges. This type of user persistent credentials is obtained in the process of the requestor (user or system) registration and/or certification. Identity credentials can be presented to and verified by the AuthN service that issues AuthN ticket or token which can be used by AuthZ service together with the requestor’s attributes received from AuthN service or obtained from Attribute Authority.

When accessing the service or resource and passing AuthN and AuthZ, the requestor can obtain temporal credentials like Proxy certificate, AuthN/Z ticket, Session ID or cookie that can be used further for identifying requestor access to the resource or ongoing session/process. These credentials are stored on the user system and their use and protection are defined by the Requestor/User client.

5 Addressing Known Vulnerabilities and Threats in Security Services Design and Operational Procedures

Presented in the previous sections the analysis and classification can be used for developing initial recommendations how to address identified Grid and Web Services vulnerabilities and threats in the general and security middleware services design and in the operational procedures.

It is understood that with wider Web Services and Grid deployment the reality and practice will bring to the surface and reveal new vulnerabilities and possible attacks but at least at this stage most of identified security concerns can be addressed in the design and operations. This is one of the goals of ongoing security coordination activity in the framework of the EGEE project and associated Middleware Security Group (MWSG) and Joint Security Policy Group (JSPG) which is also coordinated with the Open Science Grid (OSG) in US.

Most of mentioned above user and service configuration vulnerabilities (see UCA and SMA groups) can be avoided by the proper design and testing procedures at the development stage or discovered with the proper developed security auditing procedures. Operational procedures must also reflect special rules and procedures for security services deployment and management, first of all, concerning service and user credentials.

 Attacks related to malicious input and particularly attacks against XML processing system can be addressed by so-called XML Firewall which is currently available from some vendors [15, 16]. XML Firewall provides additional functions to check data authenticity, integrity and validity at the level of inspecting SOAP messages flow [16].

Proposed in the section 4 security model intends to provide a common reference model how security services should interact to provide an attack-resilient multilayer protection for Grid and Web Services.

---

2 It is suggested that at the time of the Grid2005 Workshop authors will be able to present more structured recommendations to the developers and operational services after another round of discussions in the framework of mentioned coordination groups.
6 Conclusion

Presented in the paper analysis is actually one of the first attempts to create a security vulnerabilities/treats model of interacting Web Services and Grids. All existing models are mostly concerned with the application security problems at the application side only. Proposed security model addresses security issues at all application tiers by introducing security zones for basic security services that altogether define multilayer service/resource protection.

It is intended that this analysis will create a basis for further discussion and development of more detailed security models of the Grid services in general and security services in particular. Suggested future development includes extending the security model to include distributed authentication and authorisation services, Proxy certificate management, VOMS security model, distributed policy enforcement infrastructure, etc.

Other specific topic to be targeted in the further security model development is concerned with the trust management in a dynamic policy enforcement infrastructure built around VO and/or transient Grid tasks or jobs.

Proposed security model and threats analysis can also be used for security risk evaluation in real Grid systems and as a basis for Operational procedures revision.

7 References


