EGEE

GRID AND WEB SERVICES SECURITY VULNERABILITIES AND THREATS ANALYSIS AND MODEL

Document identifier:	**draft-jra3-xws-grid-threats-analysis-02.doc**
Date:	**22/08/2005**
Activity:	**JRA3: Security**
Document status:	**DRAFT version 1-1**
Document link:	https://edms.cern.ch/document/632020/

Abstract: The document provides and overview of available Web Services security vulnerability models, which are used as a basis to create the vulnerabilities and threats model of the basic Grid middleware security services.

Document Log

Issue	Date	Comment	Author/Partner
0-1	2005-04-21	The JRA3 document on XML Web Services and Grid security vulnerabilities and threats analysis is spun off from the MJRA3.6 document	Yuri Demchenko
0-2	2005-08-22	Document updated with editorial changes	Yuri Demchenko

Document Change Record

Issue	Item	Reason for Change

CONTENT

1. INTRODUCTION

1.1. PURPOSE

This document presents an ongoing work of JRA3 that intends as its final result to provide recommendations to the security middleware developers how to address identified specific Grid security vulnerabilities, in first row, vulnerabilities of the basic security services that affect Grid applications security: authentication, authorisation, confidentiality or data protection, remote access and communication.

The document also intends to provide a bridge between operational security people and security middleware developers to support required operational security procedures by proposing common security models and approaches to addressing known security problems in both Operational Security Procedures and middleware security services implementation.

Current Grid security vulnerabilities and threats analysis is built upon the previous milestone "MJRA3.4 - Grid Security Incident definition and exchange format" [R1] and also reflects ongoing discussion in EGEE/LCG/OSG Joint Security Policy Group (JSPG). It provides further overview of existing security vulnerabilities models and classifications and proposes new security threats model and vulnerabilities classification for interacting Web Services and Grid systems that can be used for further research into Grid specific security vulnerabilities/threats. Proposed models and classification use the GridPP document on "Grid Security Vulnerability Detection and Reduction" [R2] as a valuable input and intends to provide higher level view on perceived threats and risks in Grid middleware security services and operation.

The document itself is created as a spin-off from the initial MJRA3.6 deliverable on EGEE/LCG Operational Security procedures and is specially devoted to the XML Web Services and Grid security vulnerability analysis. The document presents the overview of existing XML Web Services and Web application vulnerability analysis and proposes systemised approach and a security model for Web Services and Grid security vulnerability analysis.

1.2. APPLICATION AREA

This document intends to provide basic information for Web Services and Grid middleware developers about known and perceived security vulnerabilities caused by XML-based technologies and also dependent on possible implementation and configuration vulnerabilities. It is expected that the approach can be used for detailed vulnerabilities and risk analysis of existing middleware implementation.

1.3. REFERENCES

[This subsection provides a complete list of all documents referenced elsewhere in the document.]

[R1]	MJRA3.4 - Grid Security Incident definition and exchange format. - https://edms.cern.ch/document/501422/1
[R2]	Shirey R., "Internet Security Glossary", RFC2828. May 2000. Available at http://www.faqs.org/rfc/rfc2828.txt
[R3]	The Ten Most Critical Web Application Security Vulnerabilities. 2004 Update. - January 27th, 2004. - http://www.owasp.org/documentation/topten.html
[R4]	Enterprise Vulnerability Description Language (EVDL) v0.1. OASIS Draft, February 2005 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=was

[R5]	Web Application Security Consortium: Threat Classification, Version: 1.00, 2004. Available at http://www.webappsec.org/projects/threat/
[R6]	Improving Web Application Security: Threats and Countermeasures Roadmap, by J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan. - Microsoft Corporation. - June 2003. - 919 p. - http://msdn.microsoft.com/library/en-us/dnnetsec/html/ThreatCounter.asp
[R7]	Anatomy of a Web Services Attack: A Guide to Threats and Preventative Countermeasures - Forum Systems, Inc., March 1, 2004 - http://whitepapers.itsj.com/detail/RES/1084293354_294.html
[R8]	Attacking and Defending Web Services, A Spire Research Report, January 2004. - http://www.forumsystems.com/ papers/Attacking_and_Defending_WS.pdf
[R9]	Security Considerations for the Implementation of Unicode and Related Technology. Draft Unicode Technical Report #36. - http://www.unicode.org/reports/tr36/tr36-2.html
[R10]	Cornwal L., "GridPP Grid Security Vulnerability Detection and Reduction". - http://agenda.cern.ch/fullAgenda.php? ida=a051137
[R11]	VOMS : Virtual Organization Membership services - http://infnforge.cnaf.infn.it/voms/
[R12]	URL Security Zones, MS Internet Explorer, MSDN.- - http://msdn.microsoft.com/library/default.asp?url=/workshop/security/szone/urlzones.asp
[R13]	JSR-000154 Java™ Servlet 2.4 Specification (Final Release) - http://www.jcp.org/aboutJava/communityprocess/final/jsr053/
[R14]	Tomcat Security overview and analysis - http://www.cafesoft.com/products/cams/tomcat-security.html
[R15]	BEA Weblogic Server. Security fundamentals. - http://e-docs.bea.com/wls/docs61/security/concepts.html
[R16]	Web Services Security Firewall - http://www.forumsystems.com/products_xwall.htm
[R17]	A Guide to Securing XML and Web Services. - ZapThink, LLC - January 1, 2004 - http://whitepapers.itsj.com/detail/RES/1073404572_221.html

1.4. DOCUMENT EVOLUTION PROCEDURE

The document provides overview of the ongoing work at JRA3 on Web Services and Grid security vulnerabilities analysis and therefore will be updated as the work will progress.

1.5. TERMINOLOGY

Glossary

EGEE	the Enabling Grids for e-Science project
JSPG	Joint Security Policy Group
OSG	Open Science Grid
GSInc	Grid Security Incident

OWASP	The Open Web Application Security Project
EVDL	Enterprise Vulnerabilities Description Language
Malifactor	The person with malicious intents, e.g. intruder or attacker in the security incident.
Spoofing	A technique used to gain unauthorized access to computers, whereby the intruder sends requests indicating that the request is coming from a trusted host/site or user.

2. GENERAL APPROACH AND EXISTING WEB APPLICATIONS VULNERABILITY MODELS

This section provides short overview and summary of existing approaches to web applications security vulnerabilities analysis and modelling. This information will be used to propose high-level security vulnerability models of basic applications security services. The models are required for understanding how vulnerability may become a potential security threat and what may be possible scenarios for developing threat into attack. This will be used to identify how known vulnerabilities can be addressed in the operational security procedures and in the general requirements to middleware security services design.

The following Vulnerability-Incident life-cycle can be used as a framework for analysing relations between major components of applications and operational security:

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 [R1, R2].

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 Figure 2.1:

- Survey and assess
- Exploit and penetrate
- Escalate privileges
- Maintain access or Deny service
- Unauthorised use of Resource
- Clean or forge track of activity

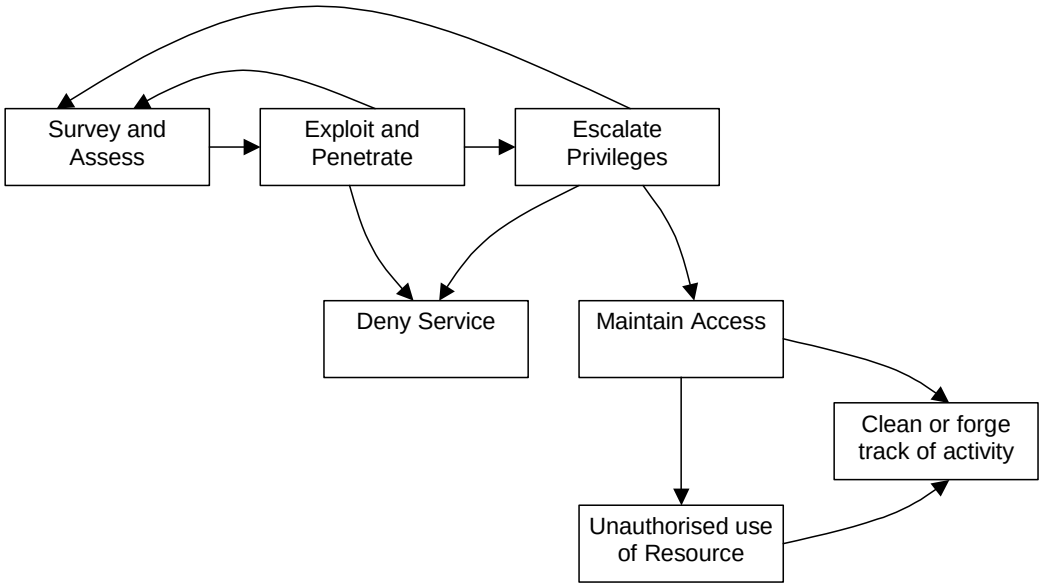

Figure 2.1. Basic steps for attacking methodology.

There are few known projects and initiatives that focus mostly on vulnerability/threat analysis and suggest general and application specific countermeasures and additionally provide recommendations for events logging that can be used for attack/intrusion identification. They are discussed below.

2.1. OWASP, WASC AND EVDL VULNERABILITIES CLASSIFICATION

The commonly recognised basic classification of the web application vulnerabilities was proposed by Open Web Applications Security Project (OWASP) in 2003-4 [R3] in their major document "OWASP 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.

A short summary of OWASP the most significant web application security vulnerabilities and their relations to major suggested XML Web Services XML-based vulnerabilities were described in the previous MJRA3.2 milestone document [R1].

Proposed by OASIS Web Application Security TC the Enterprise Vulnerability Description Language (EVDL) is also based on OWASP classification however proposing more detailed breakdown of the major vulnerabilities [R4]. 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.

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 [R5].

Table 2.1 below provides mapping between OWASP Top Ten and EVDL vulnerability categories.

Table 2.1 Mapping between OWASP and EVDL Vulnerability categories.

OWASP Vulnerabilities classification [R3]	**EVDL category**	**EVDL Vulnerabilities classification [R4]**	Comments (target and/or suggested measures)
A1 - Unvalidated Input Information from web requests is not validated before being used by a web application. Attackers can use these flaws to attack backend components through a web application.	EVDL1	InputValidation InputValidation.User InputValidation.Network InputValidation.File	Affect end service
A2- Broken Access Control Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorized functions.	EVDL2	AccessControl	Authorisation/policy enforcement configuration
A3 - Broken Authentication and Session Management Account credentials and session tokens are not properly protected. Attackers who can compromise passwords, keys, session cookies, or other tokens can defeat authentication restrictions and assume other users' identities.	EVDL3	Authentication Authentication.User Authentication.UserManagement Authentication.Entity Authentication.SessionManagement	
A4 - Cross Site Scripting (XSS) Flaws The web application can be used as a mechanism to transport an attack to an end user's browser. A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user.	EVDL4	Injection.XSS	Check input origin, source AuthN
A5 - Buffer Overflows Web application components in some languages that do not properly validate input can be crashed and, in some cases, used to take control of a process. These components can include CGI, libraries, drivers, and web application server components.	EVDL5	IntegerOverflow BufferOverflow BufferOverflow.Heap BufferOverflow.Stack BufferOverflow.Format	Affect end service, require proper end service implementation
A6 - Injection Flaws Web applications pass parameters when they access external systems or the local operating system. If an attacker can embed malicious commands in these parameters, the external system may execute those commands on behalf of the web application.	EVDL6	Injection Injection.SQL Injection.HTML Injection.OSCommand Injection.LDAP	See A4

A7 - Improper Error Handling Error conditions that occur during normal operation are not handled properly. If an attacker can cause errors to occur that the web application does not handle, they can gain detailed system information, deny service, cause security mechanisms to fail, or crash the server.	EVDL7	ErrorHandling	See A5
A8 - Insecure Storage Web applications frequently use cryptographic functions to protect information and credentials. These functions and the code to integrate them have proven difficult to code properly, frequently resulting in weak protection.	EVDL8	DataProtection DataProtection.Storage DataProtection.Transport	
A9 - Denial of Service Attackers can consume web application resources to a point where other legitimate users can no longer access or use the application. Attackers can also lock users out of their accounts or even cause the entire application to fail.	EVDL9	AppDOS AppDOS.Flood AppDOS.Lockout	Firewall protection
A10 - Insecure Configuration Management Having a strong server configuration standard is critical to a secure web application. These servers have many configuration options that affect security and are not secure out of the box.	EVDL10	ConfigurationManagement ConfigurationManagement.Administration ConfigurationManagement.Application ConfigurationManagement.Infrastructure	
	EVDL11	Cryptography Cryptography.Algorithm Cryptography.KeyManagement	*Note: EVDL Cryptography category can be also related to A10- Configuration management and partly to A8 – Insecure Storage*
	EVDL12	Monitoring Monitoring.Logging Monitoring.Detection	
	EVDL13	Concurrency	

2.2. WEB APPLICATION SECURITY THREATS MODEL AND CLASSIFICATION BY MICROSOFT

The guide by Microsoft "Improving Web Application Security: Threats and Countermeasures Roadmap" [R6] 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 across the application tiers and at multiple

layers. A weakness in any tier or layer makes the application vulnerable to attack. Figure 2.2 [R6] shows the scope of the guide and the three-layered approach that it uses: securing the network, securing the host, and securing the application.

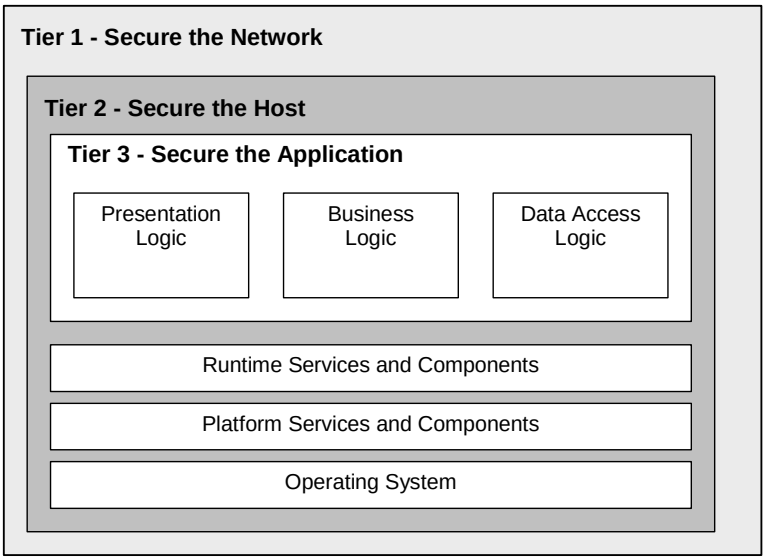

Figure 2.2. Application Security layers.

Figure 2.3 below presents major components contributing to the application security vulnerability model and countermeasures. It can also be used for the process called *threat modelling*, which provides a structure and rationale for the security process and allows to evaluate security threats and identify appropriate countermeasures at the application development stage.

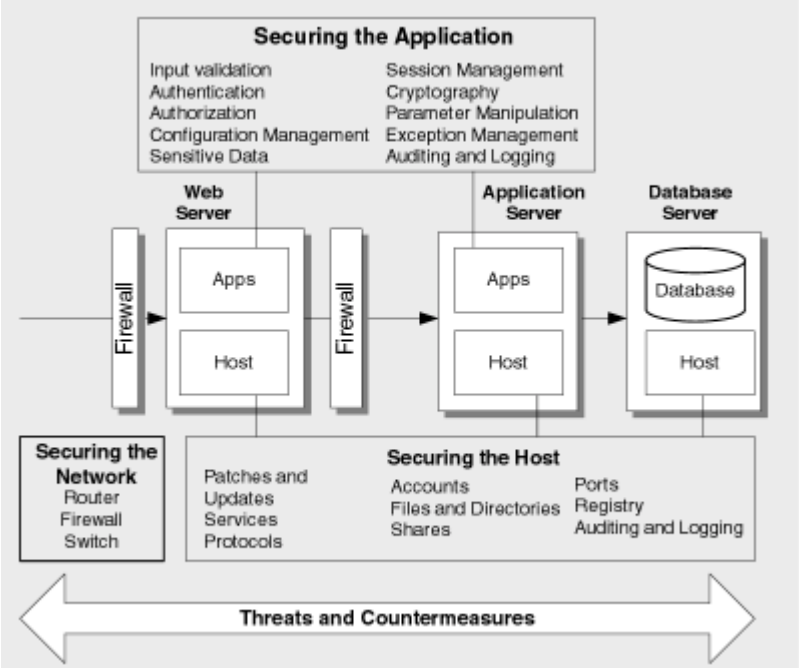

Figure 2.3. Scope of Improving Web Application Security: Threats and Countermeasures [R6].

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, however, network layer security is actually limited to node-to-node security.

Host-to-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 the (global) infrastructure 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 [R6] are categorised similar to the OWASP Top Ten vulnerabilities and provide useful recommendations for addressing them.

Table 2.2. Threats by Application Vulnerability Category

Category by service under attack (MS [R6])	Threats	OWASP mapping [R3]
SA1 – Input validation	Buffer overflow; cross-site scripting; SQL injection; canonicalization	A1 - Unvalidated Input A4 - Cross Site Scripting (XSS) Flaws A5 - Buffer Overflows
SA2 - Authentication	Network eavesdropping; brute force attacks; dictionary attacks; cookie replay; credential theft	A3 - Broken Authentication and Session Management
SA3 - Authorization	Elevation of privilege; disclosure of confidential data; data tampering; luring attacks	A2 - Broken Access Control
SA4 - Configuration management	Unauthorized access to administration interfaces; unauthorized access to configuration stores; retrieval of clear text configuration data; lack of individual accountability; over-privileged process and service accounts	A10 - Insecure Configuration Management
SA5 - Sensitive data	Access sensitive data in storage; network eavesdropping; data tampering	A8 – Insecure Storage
SA6 - Session management	Session hijacking; session replay; man in the middle	A3 – Broken Authentication and Session Management
SA7 - Cryptography	Poor key generation or key management; weak or custom encryption	A8 – Insecure Storage A10 - Insecure Configuration Management
SA8 - Parameter manipulation	Query string manipulation; form field manipulation; cookie manipulation; HTTP header manipulation	A4 - Cross Site Scripting (XSS) Flaws A6 - Injection Flaws
SA9 - Exception management	Information disclosure; denial of service	A7 - Improper Error Handling
SA10 - Auditing and logging	User denies performing an operation; attacker exploits an application without trace; attacker covers his or her tracks	

2.3. PROPOSED WEB SERVICES THREATS/ATTACKS CLASSIFICATION

This section provides short overview and further development of the proposed in the MJRA3.4 document Web Services vulnerabilities analysis and classification [R1]. The proposed classification summarises earlier works by Forum Systems and Spire Security and extends them with the potential WS-Security vulnerabilities [R7, R8].

Web Services attacks can be classified in the following way [R1]:

- 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
- 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)

- 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.)
- 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.
- 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.
- XML Security Credentials and Assertions tampering
  XML credentials and assertions are used for requestor and service authentication, authorisation and session or state management. They can be in a 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. Secure key/session negotiation tampering, Unicode content manipulation if used for credentials forging [R9]. Those can be a result of poor WS-Security implementation, poor key generation or key management; and weak or custom encryption.
- Underlying transport protocol attacks
  These attacks are not related to XML Web Services but directly affect the reliability of SOAP communications; they should be addressed at the network infrastructure level.

Table 2.3 below summarises the Web Services vulnerabilities and provides their mapping to OWASP classification [R3].

Table 2.3. Web Services threats/attacks classification

Category of threats/attacks [R1]	Threats	Suggested OWASP mapping [R3] *)
XWS1 – Web Services Interface probing	WSDL scanning, WSDL parameters tampering, WSDL error interface probing	A1 - Unvalidated Input
XWS2 – XML parsing system	Recursive XML document content, oversized XML document	A1 - Unvalidated Input
XWS3 – Malicious XML content	Malicious code exploiting known vulnerabilities in back-end applications, viruses or Trojan horse programs, malicious XPath or XQuery built-in operations, malicious Unicode content	A1 - Unvalidated Input A4 - Cross Site Scripting (XSS) Flaws
XWS4 – External reference attacks	Malicious XML Schema extensions, namespace resolution manipulation, external entity attacks	A1 - Unvalidated Input A4 - Cross Site Scripting (XSS) Flaws
XWS5 – SOAP/XML Protocol attacks	SOAP flooding attack, replay attack, routing detour, message eavesdropping, “Main-in-the-middle” attack	A3 – Broken Authentication and Session Management
XWS6 – XML security credentials tampering	XML Signature manipulation, secured XML content manipulation, Unicode content manipulation, XML credentials replay, application session hijacking	A3 – Broken Authentication and Session Management A1 - Unvalidated Input
XWS7 – Secure key/session negotiation tampering	Poor WS-Security implementation, poor key generation, poor key/trust management; weak or custom encryption	A2 – Broken Access Control A3 – Broken Authentication and Session Management A10 - Insecure Configuration Management

Note:

*) There is no direct mapping between proposed Web Services threats/vulnerabilities as OWASP and XWS classifications use different security model: OWASP describes application vulnerabilities, and XWS vulnerabilities describe interacting Web Services.

2.4. SUMMARY

Discussed above existing vulnerabilities and threats classifications and models consider only web applications or web services that provide service on user request. User/requestor site threats and “wire”/transport services is out of scope in those models. Proposed in MJRA3.4 and extended in current document the analysis of XML Web Services vulnerabilities and threats covers also specific XML related security vulnerabilities caused by using XML as a data abstraction and exchange format and SOAP/XML protocol as a transport protocol for control information (request/response) and data or service delivery.

The next section uses all these models and classifications as a basis to develop the security model of interacting Web Services and Grid services.

3. GRID SECURITY VULNERABILITIES AND THREATS ANALYSIS

The goal of this section is to provide initial analysis of possible vulnerabilities of the basic Grid security middleware services and sub-systems. First of all it will be focused on primary JR3 area of interest: Authentication, Authorisation, Credentials use and management.

This area continues to be new for researchers and developers and there is not much information available. The vulnerabilities analysis is based on existing approaches and models reviewed in the previous section and provides further development to the work done in EGEE MJRA3.4 deliverable and further abstraction to the Grid Security Vulnerabilities checklist produced by the GridPP project and discussed in MWSG [R10].

Proposed in this section security threats model for interacting Web Services and Grid systems can be considered as a first step to further creation of more detailed security models for basic security middleware services, in particular:

- Authentication system
- Authorisation system
- Credentials use and management
- Remote access and communication
- Data protection

One of goals in creation of the mentioned above security models is to define basic requirements to other middleware components such as Firewall and logging system and address known vulnerabilities in the Operational procedures.

Proposed below security threats model intends to address known vulnerabilities and concerns in current Grid middleware implementation and provide a general approach to both security design and operational security.

3.1. SECURITY THREATS MODEL FOR INTERACTING GRID SERVICES

Figure 3.1 below provides a general view and identifies major source of security threats and possible attacks related to different components and subsystems of interacting services represented by the Requestor/User and Service/Resource. The model 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[1] **Initiated Attacks.** This group of attacks can be undertaken by a potential attacker using

[1] The person with malicious intents, e.g. intruder or attacker in the security incident

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.

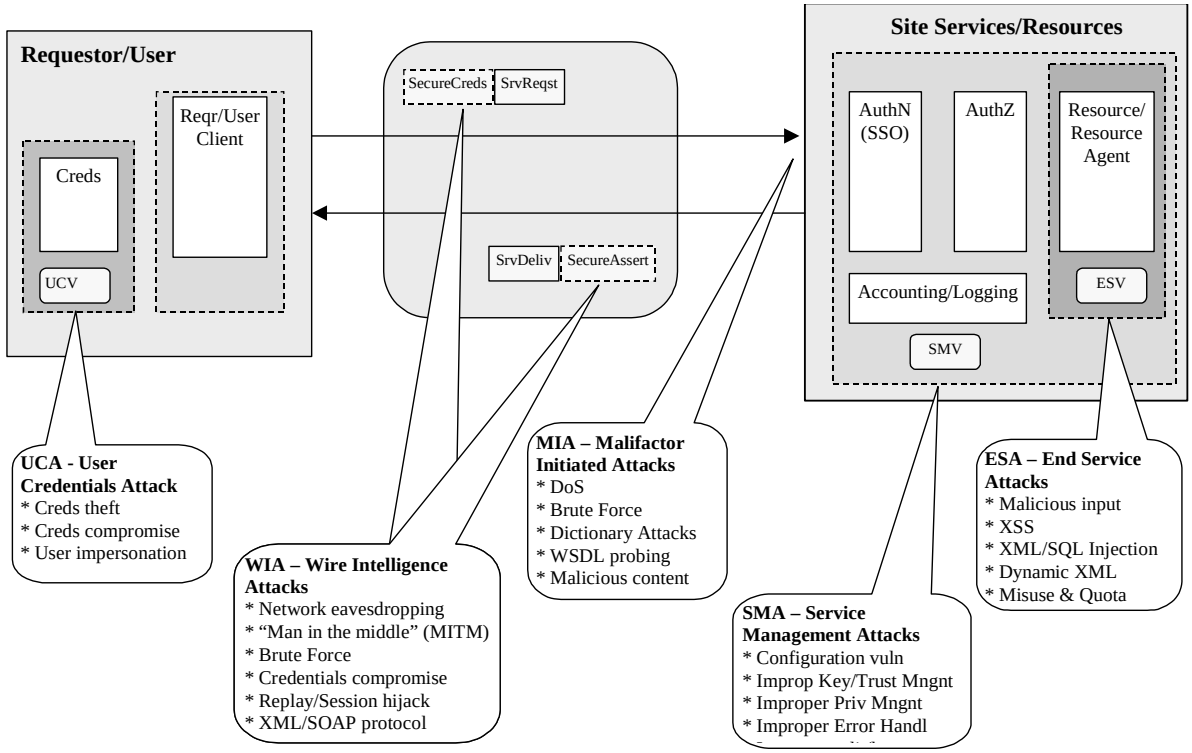

Figure 3.1. Threats/Attacks grouping in interacting services

Table 3.1 provides more detailed break-down of the identified groups and their relation and mapping to OWASP web application vulnerabilities classification and proposed in this document Web Services attacks/threats classification.

Table 3.1. Threats/Attacks groups in interacting Web Services and Grids

Threats/Attacks groups	Threats	XWS threats mapping	OWASP mapping [R3]
UCA – User Credentials Attacks	• Credentials theft • Credentials compromise • User impersonation	XWS6 – XML credentials tampering XWS7 – Secure key/session negotiation tampering	A8 – Insecure Storage A10 - Insecure Configuration Management
WIA – "Wire" Intelligence Attacks	• Network eavesdropping • "Man in the middle" (MITM) • Brute Force • Credentials compromise • Replay/Session hijack • XML/SOAP protocol	XWS5 – XML Protocol attacks XWS6 – XML credentials tampering XWS7 – Secure key/session negotiation tampering	A2 - Broken Access Control A3 - Broken Authentication and Session Management
MIA – Melifactor Initiated Attacks	• DoS • Brute Force • Dictionary Attacks • WSDL probing	XWS1 – Web Services Interface probing XWS2 – XML parsing system XWS3 – Malicious XML content	A1 - Unvalidated Input A4 - Cross Site Scripting (XSS) Flaws
SIA – Site Management Attacks	• Configuration vulnerabilities • Improper Key/Trust Management • Improper Privilege Management • Improper Error Handling • Insecure audit/log	XWS7 – Secure key/session negotiation tampering	A2 - Broken Access Control A7 - Improper Error Handling A8 – Insecure Storage A10 - Insecure Configuration Management
ESA – End Services Attacks	• Malicious input • XSS • XML/SQL Injection • Dynamic XML • Resource misuse and quota violation	XWS4 – External reference attacks XWS3 – Malicious XML content	A1 - Unvalidated Input A4 - Cross Site Scripting (XSS) Flaws A5 - Buffer Overflows A6 - Injection Flaws

4. SECURITY MODEL FOR INTERACTING WEB SERVICES AND GRID

4.1. RESOURCE/SERVICE SECURITY ZONES AND MULTILAYER ACCESS CONTROL

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) [R11].

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/assertion and policy

reference.

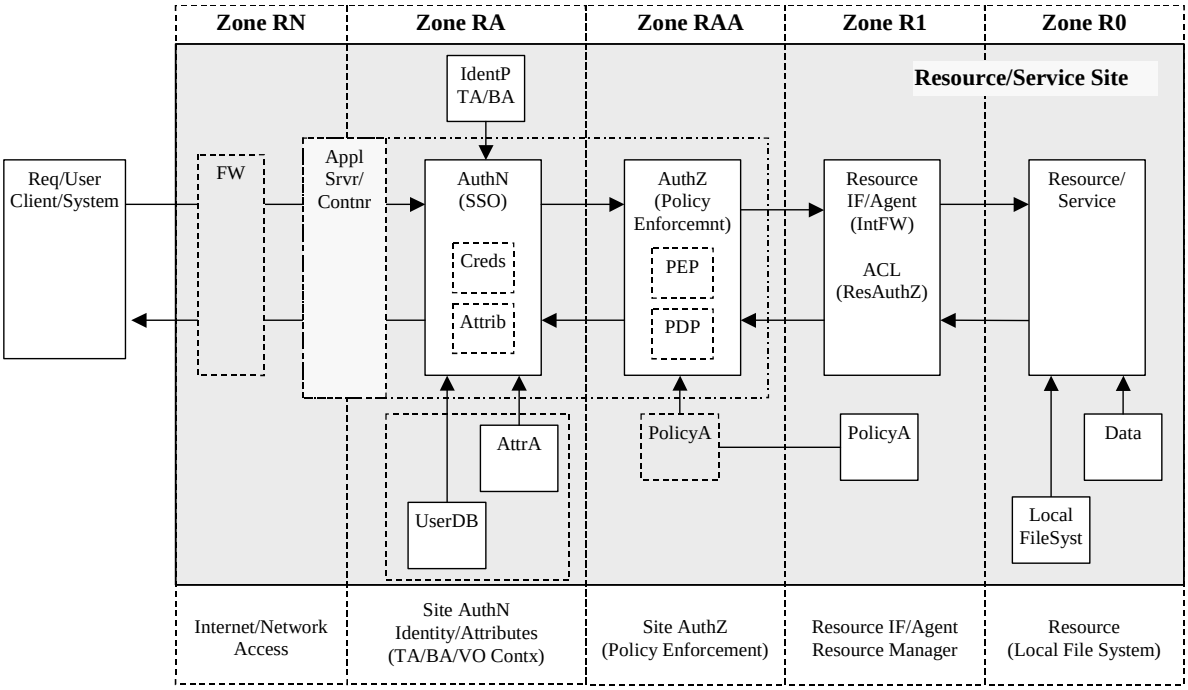

Figure 4.1. Service/Resource site security zones.

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 [R12] or security realms concept used in the Java Servlet specification [R13] and implemented in the popular servlet container Apache Jakarta Tomcat [R14], and provide better granularity required 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.

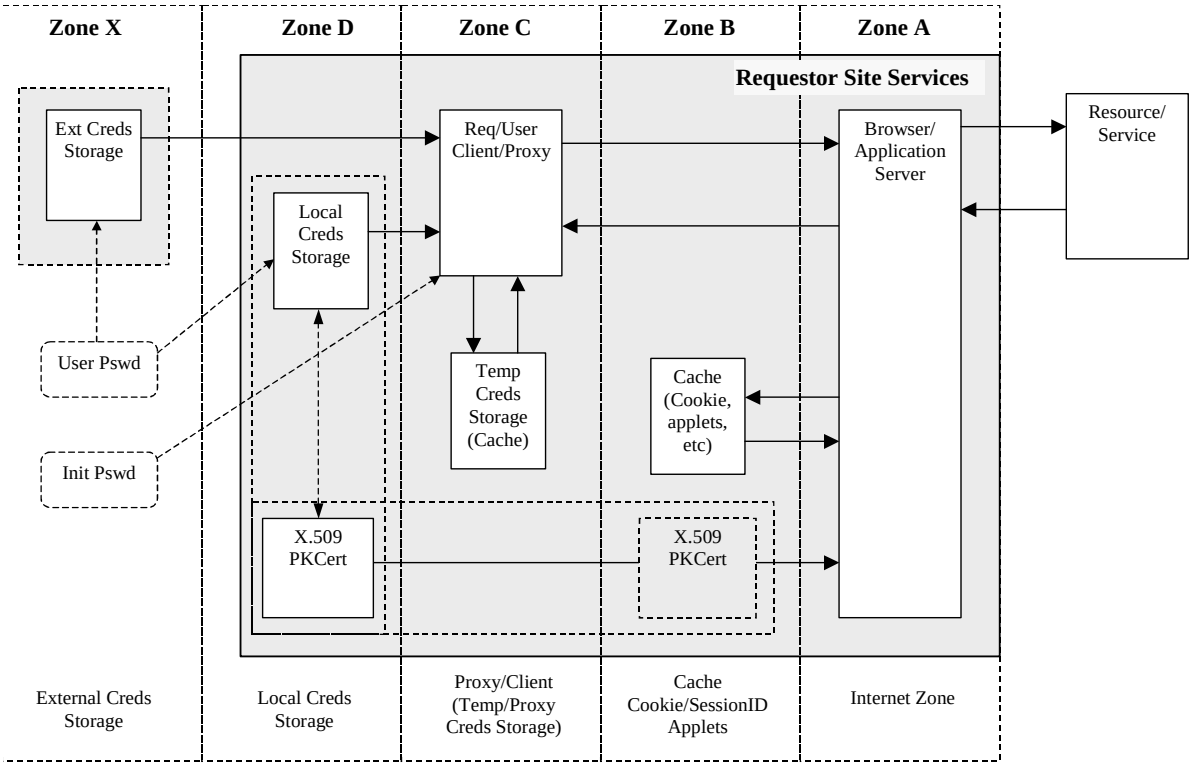

Figure 4.2. Requestor/User site security zones and credentials management.

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[2]

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 [R16, R17]. XML Firewall provides additional functions to check data authenticity, integrity and validity at the level of inspecting SOAP messages flow [R17].

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] This section will be updated as the work will progress.

6. SUMMARY AND FUTURE DEVELOPMENT

Proposed in this section analysis is actually the first attempt to create a security vulnerabilities/treats model of interacting Web Services and Grid systems. All existing models are mostly concerned with the application security problems at the application side only.

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 further developments includes: 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 relations 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.