“White collar” Attacks on Web Services and Grids
Grid Security threats analysis and Grid Security Incident data model definition

Draft Version 0.3, March 14, 2005

Yuri Demchenko <demch@science.uva.nl>

1. Goals

1) Analyse specifics of the Grid Security Incident (GSInc) based on generic Web Services threats analysis, and

2) Define general requirements to GSInc description format and suggested extensions of the emerging format for the security Incidents description IODEF

2. Grid Security Incident definition

2.1 Classical definition of Incident

1) A computer/ITC security incident is defined as any real or suspected adverse event in relation to the security of a computer or computer network. Typical security incidents within the ITC area are: a computer intrusion, a denial-of-service attack, information theft or data manipulation, etc.

An incident can be defined as a single attack or a group of attacks that can be distinguished from other attacks by the method of attack, identity of attackers, victims, sites, objectives or timing, etc.

2) An Incident in general is defined as a security event that involves a security violation. This may be an event that violates a security policy, UAP, laws and jurisdictions, etc.

A security incident may be logical, physical or organisational, for example a computer intrusion, loss of secrecy, information theft, fire or an alarm that doesn't work properly. A security incident may be caused on purpose or by accident. The latter may be if somebody forgets to lock a door or forgets to activate an access list in a router.

2.2. Incident – any specifics for Grid?

In general, Grid systems will be susceptible to all typical network and computer security threats/attacks but Grid specifics will bring new range/types of threats, first of all, inherited from XML web Services which extensive analysis is conducted in the next section.

XML/SOAP based Request messages often convey application specific commands as XML elements’ content. In this way, Grid and/or Web Services will expose all existing vulnerabilities in back-end/legacy applications and may provide a channel to bypass local vulnerabilities and viruses’ security checks for these applications. This is a challenge for both Web Services and applications developers.

Specific Security Incident definition for Grids will be based on general Security Incident definition and:

1) will depend on:

- the scope and range of the Security Policy, ULA, or SLA,

2) should be based on:

- threats analysis and vulnerabilities model
- Grid processes/workflow analysis

3) should be distinguished from incidents related to the underlying networking infrastructure.

3. XML Web Services threats analysis

3.1. Generic XML Web Services threats/attacks classes

XML Web Services threats/attacks can be classified in the following way:

- Web Service interface (WSDL) probing
  WSDL as an advertising mechanism for web services describes the methods and parameters used to access a specific Web Services, and in this way exposes Web Service to possible attacks
- Brute force attack on XML parsing system
  XML parsing is a resource and time consuming process. Many real world applications may allow complex or voluminous XML files what may overload XML parsing system
- Malicious Content
  XML documents may contain malicious parsing or processing instructions (XML Schema extensions, XPath or XQuery instructions, XSLT instructions, etc) that may alter XML parsing process, or malicious content that may carry threats to the back-end applications or hosting environment (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 is based on the generic ability of XML to include references to external documents or data types. Poor configuration, or improper use of external resources can be readily exploited by hackers to create DoS scenarios or information theft.
- SOAP/XML Protocol attacks
  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.
- Underlying transport protocol attacks
  These are actually not related to XML Web Services but directly affecting reliability of SOAP communications.

3.2. Web Services interface (WSDL) probing

1) WSDL Scanning

Web Services Description Language (WSDL) as an advertising mechanism for web services describes the methods and parameters used to access a specific Web Services, and in this way exposes Web Services to possible attacks. In addition, the information provided in a WSDL file may allow an attacker to guess at other methods. For example, a service that offers stock quoting and trading services may advertise query methods like requestStockQuote, however also includes an unpublished transactional method such as tradeStockQuote. It is simple for a persistent hacker to cycle thru method string combinations in order to discover unintentionally related or unpublished application programming interfaces. Another possible scenario to overpass AuthN/AuthZ system can be derived from the analysis of the security tokens exchange by using replay techniques (this kind of exploits is also related to the next Parameter tampering threat).

Additional threat is imposed by automatic generation of the access code without checking WSDL for possible exploits that may be inserted into requestor's system.

2) WSDL Parameter Tampering

Parameters are used to convey client-specific information to the Web service in order to execute a specific remote operation. Since instructions on how to use parameters are explicitly described within a WSDL document, malicious users can play around with different parameter options in order to retrieve unauthorized internal system information, gain unauthorised access, or bypass security checks. For example, by probing Web Service with specially constructed messages an Attacker can receive error messages from the different components of XML Request processing system and guess on the internal systems structure. By submitting special characters or unexpected content to the Web service can cause a denial of service condition or illegal access to protected resources. An attacker can embed, for example, command line code into a document that is parsed by an application that can create a command shell to execute the command.

3.3. Attacks on XML parsing system

Attacks on XML parsing system are also called *Coercive Parsing*. XML processing software is a necessary component of the native XML Web Services applications or Web Services enabling middleware connecting non-XML legacy applications. This component of Web Services applications is susceptible to XML based attacks whose main objective is either to overwhelm the processing capabilities of the system or install malicious mobile code.

3) Recursive XML document (payload) content

One of the strengths of XML is its ability to nest elements within a document to address the need for complex relationships among elements. The value is easy to see with forms that have a form with many different elements, such as a purchase order that incorporates shipping and billing addresses as well as various items and quantities ordered. XML documents (and consequently XML Schema) providing this possibility normally will allow multiple elements and recursive nesting which are generally not limited in number and depth. An attacker can easily create a document that attempts to stress and break an XML parser that contains 10,000 or 100,000 elements with complex hierarchy of nested elements.

4) Oversized XML documents/payloads

XML can wrap up any type of data including multimedia or binary data. It is verbose by design in its markup of existing data and information. File size limits must be setup high (up to hundreds of megabytes or gigabytes in size) or not limited at all. So, it gives an attacker a possibility to execute a denial-of-service attack by overloading parser. Parsers based on the DOM model are especially susceptible to this attack given its need to model the entire document in memory prior to parsing

3.4. Malicious XML Content

5) Malicious code exploiting known vulnerabilities in applications

In many implementations XML/SOAP messages transfer command calls to back-end applications. In these cases, malicious code conveyed as XML content can target such vulnerabilities as buffer overflow, Unicode based vulnerabilities, etc.

6) Viruses, or Trojan horse programs

Legacy or just non-XML applications using Web Service front-end interface can be exposed to the same attacks as in direct access. Viruses, or Trojan horse programs, being transmitted within otherwise valid XML messages can bypass normal virus scan protecting an application from direct attacks. Binary attachments such as images, executables, and application-specific documents can all be modified to cause exceptions within the Web Service application.

7) Malicious XPath or XQuery built-in operations

Often XML documents use rich XPath or XQuery instructions format to define some required operations on the content. Such operations can combine few components into one, alter another content before sending it to application what may allow for malicious code or content to bypass direct security checks. Theoretically, this exploit may be used to manipulate security tokens inside one document between legitimate/authorized content and unauthorized or malicious one.

8) SQL Injection

Database front-end XML parsers are aimed at native database languages in the same fashion as SQL injection. SQL injection could allow an attacker to execute multiple commands in an input field by using native command separators like ‘;’ or pipes. This capability may allow an attacker to illegitimacy retrieve, update or insert information in the database.

3.5. External Reference Attacks

This group of threats can be also added to the malicious content group but for further Intrusion prevention analysis it is better to be grouped separately. This group is based on the generic ability of XML to include references to external documents or data types.

These vulnerabilities can be readily exploited by hackers to create DoS scenarios, information theft, or more general system misuse.

9) Malicious XML Schema extensions (Schema Poisoning)

XML Schemas provide formatting instructions for parsers when interpreting XML documents. XML Schema can reference external data types by including reference to external Schemas or namespaces. This versatility of Schema makes it susceptible to poisoning. An attacker may attempt to compromise the schema in its stored location and replace it with a similar but modified one.

Denial-of-service attacks against the grammar are straightforward if the schema is compromised. In addition, the door is open to manipulate data if data types are compromised, like modifying the encoding to allow for data mimicking that eventually gets through to a parser and re-formed into an attack. In the same way, when using extensive range of data formats (first of all, Unicode or multilingual data), XML schema may reference external transformation methods referenced by URL, which may be tampered or substituted with a malicious code.

10) External Entity Attack

Benefit of XML in ability to build documents dynamically at the time of parsing or composing by pointing to a URI where the actual data exists may expose a service to non-trustworthy external entities. An attacker can then replace the data being collected with malicious data.

3.6. XML Protocol threats/attacks

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.

11) SOAP Flooding Attack (DoS)

A hacker can issue repetitive SOAP message requests in an attempt to overload a Web service. This type of network activity will not be detected as a network intrusion because the source IP is valid, the network packet behavior is valid and the HTTP request is well formed. However, the business behavior is not legitimate and constitutes an XML-based intrusion. In the replay variant of this kind of attack, a completely valid XML payloads can be used to issue a denial of service attack.

12) Replay Attacks

Replay technique may be used for both DoS attacks and a kind of “man-in-the-middle” attacks. Replay technique can also be to manipulate AuthN/AuthZ security tokens, to fraud accounting system and bypass credit limits.

13) Routing Detours

In a distributed Web Services environment SOAP messages may pass multiple intermediate systems and may be actively routed depending resource availability at specific location. The WS-Routing specification provides a way to direct XML traffic through a complex environment. It operates by allowing an interim station to assign routing instructions to a SOAP message/document. If one of intermediate stations is compromised, it may be used for a man-in-the-middle attack by inserting bogus routing instructions to point a confidential document to a malicious location. From that location, then, it may be possible to forward on the document, after stripping out the malicious instructions, to its original destination.

14) Message eavesdropping

Eavesdropping is possible in not completely secure network. Eavesdropping can gather wide spectrum of sensitive information that may be used later for launching an attack. Even if the SOAP messages content is encrypted, a lot of information can be obtained by analyzing SOAPHeaders, WSDL ports, Certtificate chain or CA trust relations, service names and addresses, etc..

15) "Man-in-the-middle" attack

One particular case of eavesdropping based attack is the "man-in-the-middle" attack that may target any subsystem of the target system. One specific type of attack that may be ultimately based on "man-in-the-middle" method is an attack on cryptographic system or related security services, for example, private key compromise, credentials theft or compromise, AuthN/AuthZ tokens tampering, etc.

3.7. Underlying transport protocol attacks

These are actually not related to XML Web Services but having direct effect on SOAP messaging performance and availability. Threats can include HTTP or HTTPS DoS attacks, or even lower layer transport level attacks.

4. Grid specific risks and threats

4.1. Grid security risks analysis

First Grid risks analysis from the operational point of view in made in the LCG project (see LCG Risk Analysis – http://proj-lcg-security.web.cern.ch/proj-lcg-security/RiskAnalysis/risk.html).

Proposed basic classification of risks (from the operational point of view):

1) Misuse
2) Confidentiality and Data integrity
3) Infrastructure disruption
4) Accidental categories

can be extended with more technology dependent

5) XML Web Services vulnerabilities/risks based on the analysis in the previous section

4.2 Grid processes/workflow analysis

TODO

LCG definition of the Grid Job/Task submission:

Job submission will normally progress from a User Interface (UI) machine, through a Resource Broker (RB) to a Computing Element (CE) and hence to the compute resource (usually a batch system). In some cases the RB is not used and the UI submits the job directly to the CE. Data access is through a Storage Element (SE) service

4.3 Analysis of the Security Policy and ULA/SLA for typical Grid applications

TODO

5. Protecting Grid and Web Services against known threats and vulnerabilities

TODO

1) Message alteration and eavesdropping

Message alteration and eavesdropping can be addressed by using the integrity and confidentiality mechanisms described in WS-Security.

2) Replay attacks

Replay attacks can be addressed by using message timestamps and caching, as well as other application-specific tracking mechanisms.

3) man-in-the-middle attacks

For WS-Security and SAML assertion tokens whose ownership is verified by use of keys, man-in-the-middle attacks are generally mitigated by the use of subject confirmation.

It is strongly RECOMMENDED that all relevant and immutable message data be signed.

It should be noted that transport-level security MAY be used to protect the message and the security token.

6. Grid Security Incidents – known cases and analysis

Known analyses of Grid Security Incidents nature mostly focus on vulnerabilities of AuthN/Z and Certificate compromise.

Dane Skow's "A walk through a Grid Security Incident"
http://www.triumf.ca/hepix2003/pres/23-10/dskow/A%20walk%20through%20a%20Grid%20Security%20Incident-v2.ppt

Summary of UF/IU security incident – June 2004 - http://www-mcs.ivdgl.org/mail_archive/grid3-all/2004/06/msg00048.html

Some typical/perceived Grid Security Incidents are discussed below.

6.1 Private key compromise

Evidence and log/audit events:

- patterns of key usage
- broken chain of PKC/keys/credentials
- copy is discovered in not a proper place

Problem is still remaining how to define at early stage that private key has been compromised?

6.2. Other/general credentials compromise

Evidence and log/audit events:

- patterns of key usage
- broken chain of PKC/keys/credentials
- copy is discovered in not a proper place
- originated not from default location
- sequent fault attempt to do action(s)

-> PDP/PEP logging/audit

6.3. Attempt to access sensitive data/information with lower level of privileges

Evidence and log/audit events:

TODO

6.4. Credit limit on resource exhausted

Evidence and log/audit events:

- few unsuccessful attempts to run actions with unmatched credit

7. Incident Response and Intrusion Detection

Intrusion Detection (ID) and Incident Response (IR) are different components of the Operational Security framework:

* ID is rather proactive service; Incident Response is a reactive function.

* ID produces alerts to prevent suspected activity escalation to incident

* ID reacts on security events; security event may be escalated to the security incident

* ID/Network protection is a responsibility of Network Operator or Team – to be defined by SLA and IResp agreement

* CSIRT often has an influence on network security policy and IDS policy/criteria

7.1 Intrusion Detection

Intrusion Detection normally is a component of the network infrastructure/services.

Intrusion Detection Systems (IDS) or Sensors are installed on or close to Firewalls, Routers, Switches or run as a special program on logfiles.

7.2 Incident Response

Incident Response is a complex of designated people, policies and procedures. Much of Incident Response practice among CSIRTs is defined by RFC2350.

7.2.1 Incident Response Policy

Incident Response Policy includes the following components:

- Types of incidents and level of support
    - ordered by severity list of Incident categories

- Co-operation, interaction and disclosure of information
    - Based on organisation’s Security Policy
    - Availability of information and ordered list of information being considered for release both personal and vendor’s

- Communication and Authentication
    - Information protection during communication
    - Mutual authentication between communicating parties

- Also depending on information category

7.2.2. Incident Response Procedures

Should be documented in full or in critical parts

1. Initial Incident Reporting and Assessment
2. Progress Recording

3. Identification and Analysis
4. Notification – initial and in the progress
5. Escalation – by Incident type or service level
6. Containment
7. Evidence collection
8. Removal and Recovery

7.3 Grid Security Incident vs Grid Security Event

1) few sequent failed logins

2) credit limit probing

3) attempt to access sensitive information

4) SOAP port scanning

5) HTTPS DoS attack?

6) patterns of suspected private key compromise

7) patterns of suspected AuthN/AuthZ security tokens compromise

8. Using IODEF for Grid Security Incident description

8.1. IODEF and Incident Handling Framework

IODEF (Incident Object Description and Exchange Format) [R8] is a standard used by CSIRTs world wide to exchange incidents information. IODEF is compatible with another format for Intrusion Detection Systems (IDS) the Intrusion Detection Message Exchange Format (IDMEF) [R12].

This section will explain how is IODEF is used for incident reporting and handling and provides suggestions for IODEF use for Grid Security Incidents description.

Importance of adopting one of standard format for Grid Security Incidents reporting is explained by the needs of Grid Operational Security services to cooperate with external CSIRTs in incident responses activities.

8.2. Incident Reporting Format Requirements overview

This section provides overview of the recent Requirements for the Format for INcident information Exchange (FINE) [R8] produced by the IETF INCH-WG, which is considered relevant to the Grid Security Incident reporting and exchange format.

The requirement document defines the high-level functional requirements for a transport format to exchange incident reports, including general requirements to format, content, security, and related requirements to Incident Handling Systems. This abstract data representation is specified in another INCH-WG document IODEF Data model [R9].

The intent of FINE is to decrease the response time to incidents and facilitate by improving the ability of CSIRTs to process incident reports. The definition of a well-defined format will facilitate the exchange of incident reports across organizations, regions and countries by achieving these particular goals:

- to make the semantics of the report as clear and unambiguous;
- to ensure that the data has a well defined syntax;
- to ensure that the structure of the report allows easy categorization and statistical analysis;
- to ensure the verifiability of the integrity of the report, and the authenticity of the report source.

8.2.1. The Incident Reporting Operational Model

The FINE requirement draft describes the basic Incident response operational model (see picture 3.3.2.1 below).

Incident reports are generated, received and updated . For example, an organization may send an incident report to a Computer Security Incident Response Team (CSIRT) when an attack is detected. CSIRTs receive incident reports from customers or from other CSIRTs. The CSIRTs maintain these reports in an Incident Report Database in some format that may be specific to the CSIRT. The CSIRTs may process the reports to generate statistics, or investigate an incident further. As part of the investigation or as part of the reporting, the CSIRT may forward the incident report or parts of it to other CSIRTs. The CSIRTs may also receive results of investigation, or additional information related to currently active incidents from other CSIRTs. In the context of FINE, the incident reports will be handled by a CSIRT via an interface that is capable of converting a FINE formatted incident report into the internal format used by the CSIRT and vice versa.

```
               CSIRT
+------------------+                          +------------------+
|                  |                          |                  |
| +--------+   +---------+              +---------+   +--------+ |
| |        |<--|Interface|<--Incident-->|Interface|-->|        | |
| |Incident|   +---------+    Report    +---------+   |Incident| |
| | Report |       |                          |       | Report | |
| |Database|       |     |===  FINE  ===|     |       |Database| |
| |        |       |                          |       |        | |
| +--------+       |                          |       +--------+ |
|                  |                          |                  |
+------------------+                          +------------------+
```

Fig. 3.3.2.1. Operational Model for FINE

From the operational point of view during the life-cycle of an incident report the following may apply:

- the report itself evolves. It may exist in one of the following states:
  - handling - the incident report is being handled
  - complete/closed - the incident report has been processed
  - and no further processing is planned
  - waiting - the incident report is waiting on some event;
- the report is exchanged between CSIRTs and may be investigated/processed by multiple CSIRTs, simultaneously;
- additions and/or changes to the report may be made by one or more CSIRTs. Therefore, a single CSIRT may not be in a position to vouch for the veracity of all parts of the incident report.

8.2.2. General Format Requirements

The following general Incident reporting format requirements are specified:

- FINE SHALL support full internationalization and localization and use of multiple languages.
  A significant part of the incident report will be comprised of human readable text. Since some incidents will entail involvement of CSIRTs from different countries and geographic regions, FINE must have provisions for using local character sets and encodings.
  In cases where local (non-standard) character sets and encodings are used, the elements that carry encoding sensitive information should be clearly indicated. It should be possible to preserve the content of these elements when transferring an incident report.

- FINE MUST be able to document the evolution of an incident.
  An incident report may evolve with time, as further investigation is performed on the incident report. Earlier information may be modified and new information may be added. FINE must support the recording of these changes.

- FINE MUST support specifying a granular access restriction policy for the specific elements of the incident report.
  Various parts of an incident report will have information of varying degrees of sensitivity and will need to be handled with the appropriate level of confidentiality. It must be possible to specify the degree of confidentiality for the individual components of the incident report. Applications can then implement different levels of access restrictions for the different components of the incident Report.

8.2.3. Incident Report Content Requirements

FINE specifies the following requirements to the Incident report content which are also applicable for the Grid Security Incident description:

1. FINE MUST support globally unique identifiers for each incident report.
   It should be possible to reference an incident report unambiguously using a globally unique identifier. It should be possible to derive the creator of the incident report from this identifier.
2. FINE MUST include the identity of the creator of the incident report.
   FINE should indicate the source of each component of the incident report if is different from the creator (e.g., the team handling the incident).
3. FINE MUST be flexible enough to support various degrees of completeness, while still clearly defining the minimal information required for describing an incident.
   FINE SHOULD support the including or referencing information external to the incident report.
4. FINE SHOULD support the description of various aspects of the source and target.
5. FINE SHOULD contain a description of the methodology used in the attacker.
   Well-known classifications or enumeration schemes should be used to describe the attack or exploited vulnerabilities that caused the incident.
   FINE SHOULD support references to the appropriate advisories from coordination and analysis centers.
6. FINE SHOULD provide for describing the impact of the incident report.
7. FINE SHOULD support describing the actions taken during the course of handling an incident.

8.2.4. Adopting FINE Incident Reporting Format for Grid

FINE requirements are enough general to provide a guidance for various application areas including Computer Grids and XML Web Services in general. Based on overview and initial analysis above, it can be suggested that the FINE requirements can be adopted in general for Grid Security Incidents reporting.

However, this new area of use can provide even better integration between Incident reporting facilities and applications and their monitoring and logging system because of XML based nature of interactions in Web Services and Grids.

8.3. IODEF Data model status update (version 0.4)

IODEF is the product of the IETF INCH-WG which combines efforts of CSIRT community worldwide to define a common standard for security incidents description and exchange format.
Recent version of the IODEF Data model is version 0.4 which updates the last published draft on version 0.3 with new features that were discussed at the last INCH-WG meeting on March 9, 2005 at IETF-63.

The following improvements were proposed and design issues have been resolved [R11]:

- Version 0.4 signifies moving from XML DTD for IODEF Data model definition to XML Schema what intends to simplify extensions management and adding XML based security features.
- “iodef” namespace is introduced for the major IODEF datamodel.
- Design suggestions and recommendations for using XML Signature and XML Encryption were provided.
- Definition of name related elements in the Contact element has been updated
- Content of the System element that describes the system(s) involved in the incident is updated and extended with the User element and the OperationSystem element.
  Note: This is the place where the XMLWebService element is included.
- Semantics of the RecordData element that contains important collected data related to the incident is specified to allow use of rare log data, system files or other rare evidence information.

IODEF extension for Grid Security Incidents descriptions are proposed in a form of IODEF special profile for XML Web Services and Grids with the namespace “iodef-xws” discussed in details below. The work on IODEF-XWS profile definition is directly associated with the JRA3 operation security activity for EGEE.

Extending IODEF adoption as a common basic format for security incidents description is indicated by two another proposed IODEF profiles for Real-Time Internet Defence (RID) and for phishing [R11].

8.4. IODEF Structure and top level elements

This section provides the update for the basic IODEF data model and provides information about the IODEF top-level elements. This is also an update of the related section of the MJRA3.4 document [R1].

The IODEF is designed to represent all necessary information about the computer security incident during its whole lifetime in a structured way using XML. The following element definitions are provided according to the recently updated IODEF data model specification version 0.4 [8, 11].

The top level element IODEF-Document serves as a container for only one element: Incident. The Incident element contains all the incident-related information. It provides a standardized representation for commonly exchanged incident data and associates a unique identifier with the described activity.

The Incident element contains the following sub-elements and has the following structure as represented in the XML DTD format:

```
<!ELEMENT Incident (IncidentID, AlternativeID?, RelatedActivity?,
Description*, Contact+, ReportTime, DetectTime?, StartTime?, EndTime?,
EventData*, Method*, Expectation*, Assessment+, History?, AdditionalData*)>
```

IncidentID - an incident tracking number assigned to the incident by the party that generated the document.

AlternativeID - a list of incident tracking numbers used by other CSIRTs to refer to the same activity as described in the document.

RelatedActivity - a list of incident tracking numbers referencing related incidents.
Description - a free-form textual description of the incident activity.
Contact - contact information for the parties involved in the incident.
DetectTime, **StartTime**, **EndTime**, **ReportTime** - time information when the incident activity was correspondently first detected, started, ended, reported.
EventData - details on the data on the (security) events that lead to the incident.
Method - the techniques (e.g., tools, vulnerabilities) used by the attacker.
Expectation - expected action to be performed by the recipient (CSIRT) of the document.
Assessment - a characterization of the impact of the incident activity.
History - documents significant events or actions that occurred during the course of handling the incident.
AdditionalData - extension area for data that cannot be represented anywhere else.

Graphically the top level IODEF structure can be represented in a form of diagram Fig. 3.3.1.

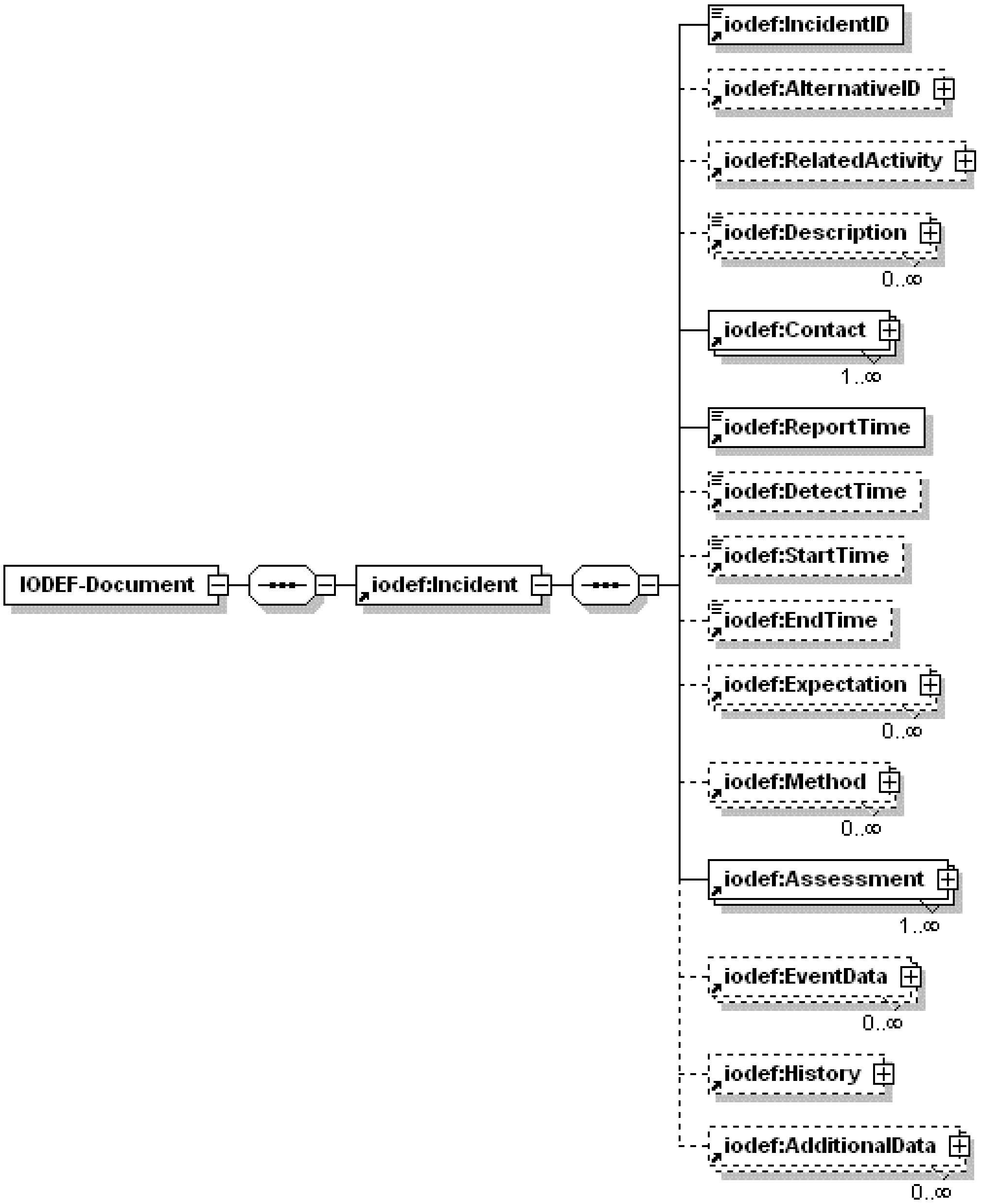


Fig. 3.3.1. The top level IODEF elements.

The Incident element may contain “zero” or “many” EventData elements where the actual incident data are contained. The EventData element has the following structure (see also Fig. 5.2):

```
<!ELEMENT EventData (Description*, Contact*, DetectTime?, StartTime?,
EndTime?, Flow*, System*, Method*, Assessment?, EventData*, Record?,
AdditionalData*)>
```

where most of the components are the same as in the higher level Incident element and two new elements are defined:

Flow – represents set/collection of similar events that may originate from the same of different Systems and applications.
System - the systems (nodes, networks) involved in the event as either sources, targets or intermediaries.
Record - support data (e.g., log files) that provides information about the events.

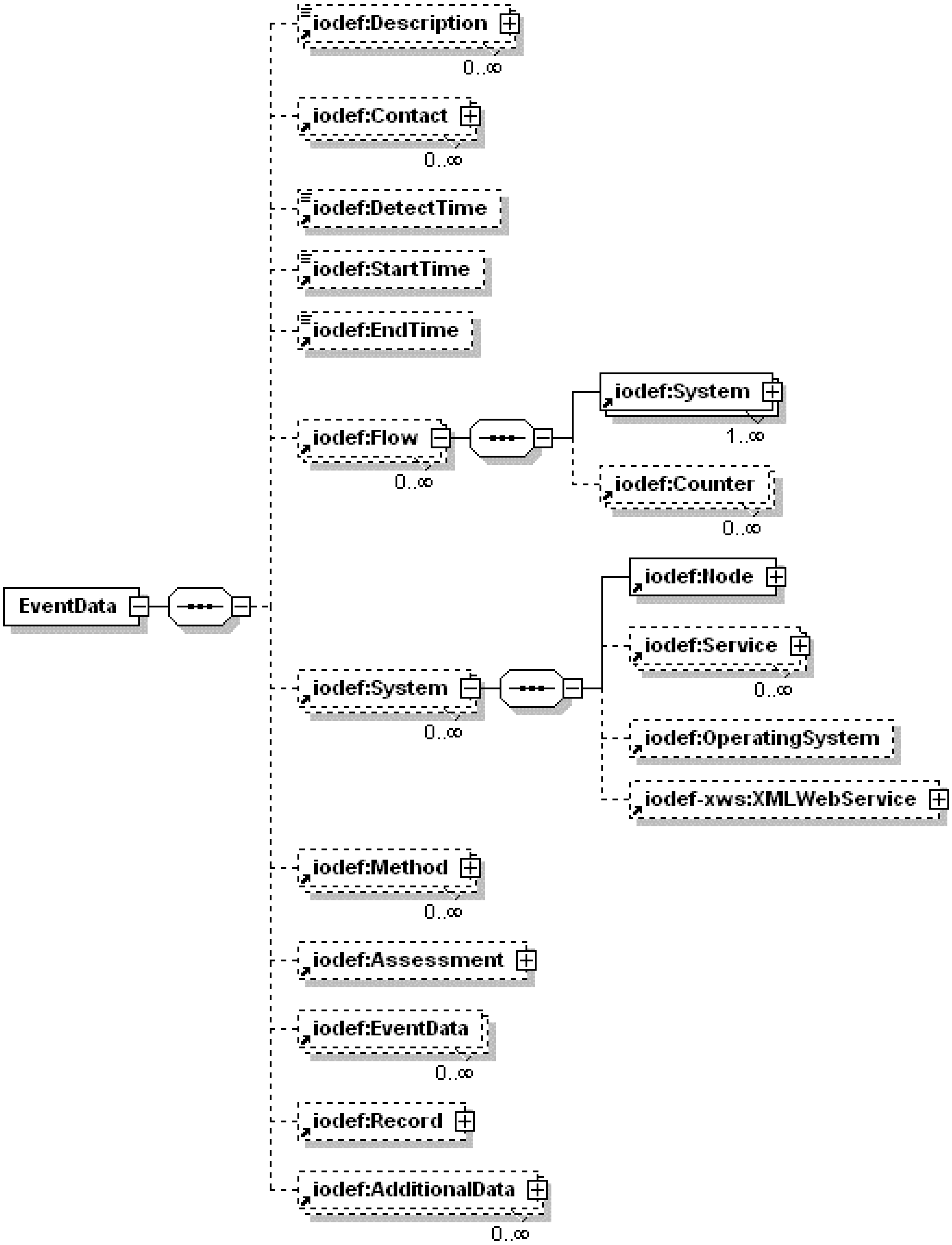


Fig. 3.3.2. The EventData element structure

The Record element may contain one or more RecordData elements that have the following structure (see also Fig. 3.3.3):

```
<!ELEMENT RecordData (DateTime?, Description*, Sensor?, Pattern?,
PatternLocation*, Counter?, RecordItem?)>
<!ATTLIST RecordData
      restriction NMTOKEN #IMPLIED
      recsourcetype CDATA #IMPLIED
>
```

DateTime - timestamp information for the RecordItem data.
Description - free-form textual description of the provided RecordItem data. At minimum, this description should convey the significance of the provided RecordItem data.
Sensor - information about the Sensor as a source of the data contained in the RecordData element. In particular case it can be Intrusion Detection System (IDS) or other intelligent event analyser. Other sources of the RecordData can be identified by the “recsourcetype” attribute.
RecordItem - log, audit, or forensic data.
Pattern – a pattern in the RecordItem data that identifies a specific incident signature.
PatternLocation – information that points on the location of pattern in the RecordItem data.
Counter – a number of pattern occurrence in the RecordItem data.

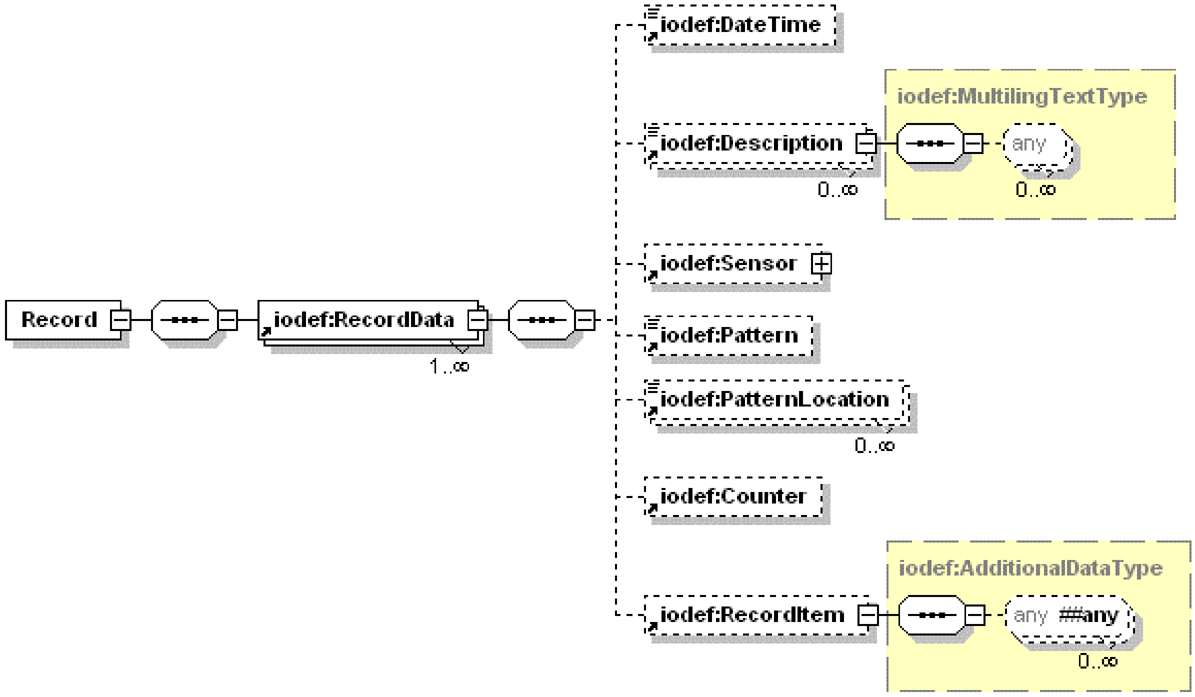


Fig. 3.3.3. The Record and RecordData elements structure

The System element contains the following sub-elements (see also Fig. 5.2):

```
<!ELEMENT System (Node, Service*, OperatingSystem*, XMLWebService*)>
```

Node - a host or network involved in the incident activity.
Service - the network service targeted on the host specified in Node.
OperatingSystem – information about the operating system installed on the Node or running the Service.
XMLWebService – description of the WML Web Service or Grid System in particular that was involved in the incident.

The elements XMLWebService is added as a child element to the System element to describe information specific for the Grid security incidents, they are described in detail in the next section.

8.5. IODEF extensions for Grid and Web Services

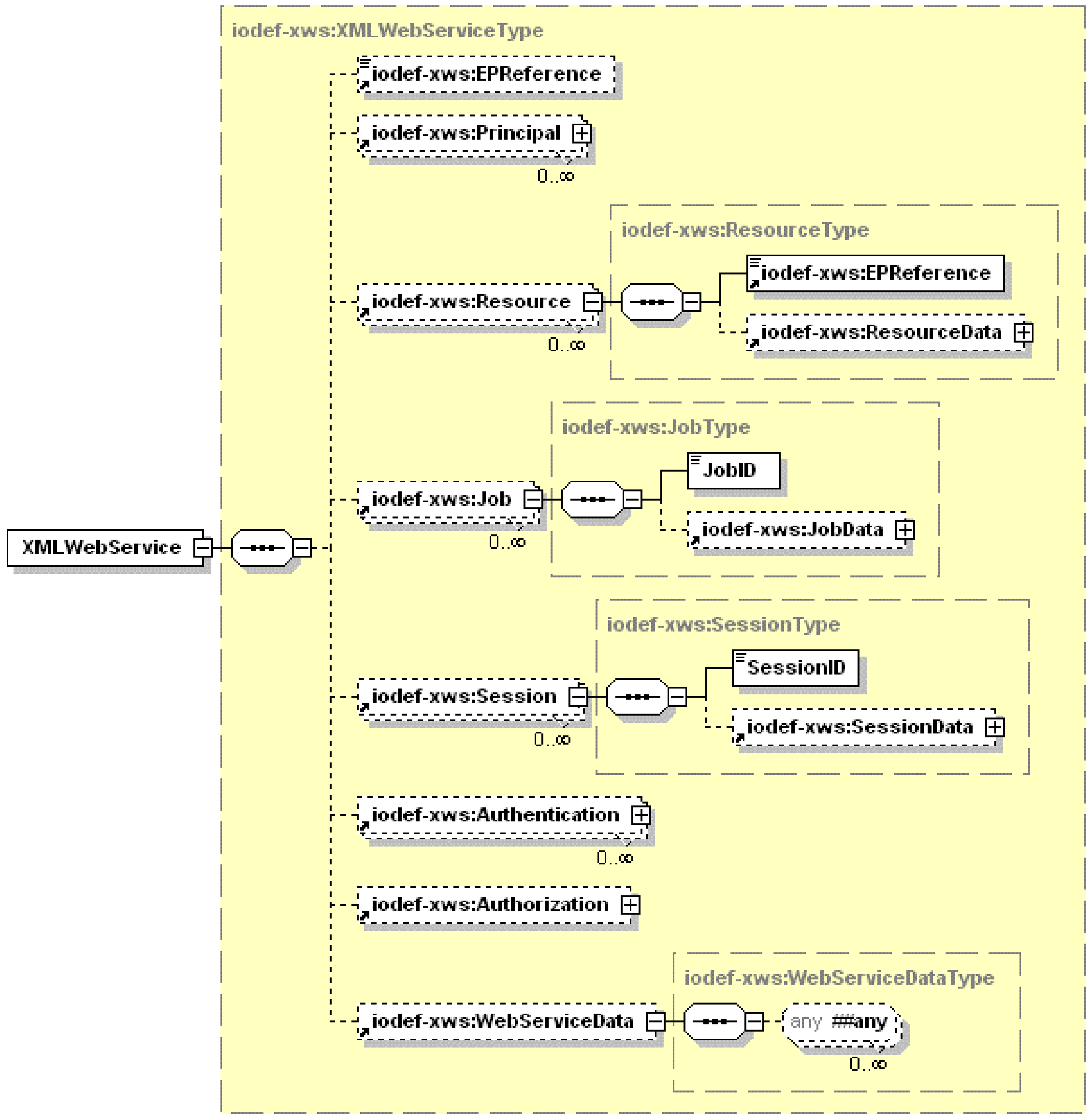
Describes and explains Grid specific IODEF elements. This is an update of the MJRA3.4 document [R1].

The proposed extension element XMLWebService and its components are described in detail in the XML Schema format below:

```
<!ELEMENT XMLWebService (EPReference?, Principal*, Resource*, Job*,
Authentication*, Authorization?, WebServiceData?)>
```

Element **XMLWebService** contains information about the Web Service or web application that was involved in the incident.

<table>
<tr>
<td>diagram</td>
<td>



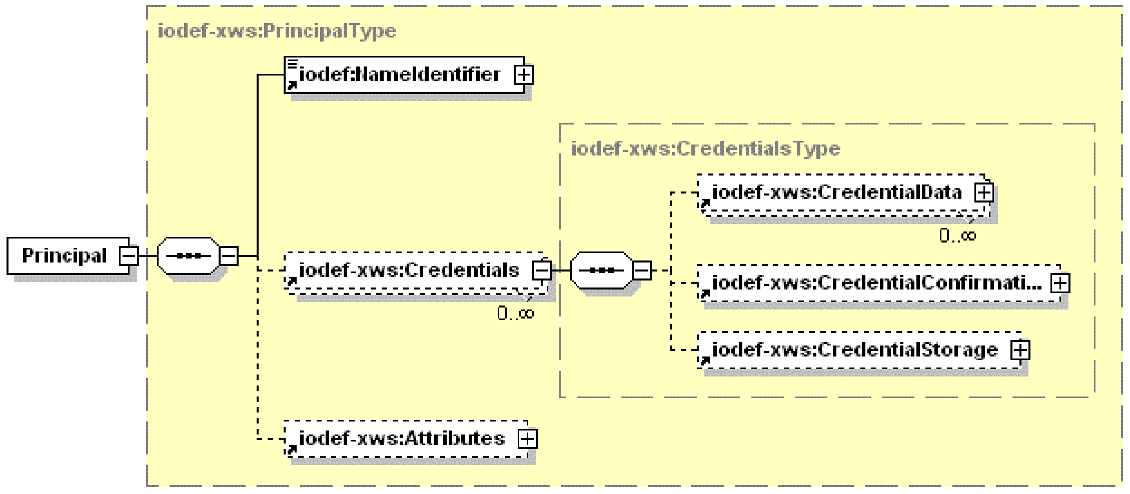
</td>
</tr>
<tr>
<td>children</td>
<td>**EPReference** – End-Point Reference that uniquely identifies Web Service or Grid Service
Principal – entity participating in Web Service request or consumption
Resource – provides information about the Resource that was involved in the incident, suggested resources may be Computer Element (CE), Storage Element (SE), etc.
ResourceData – sub-element of the Resource containing arbitrary data about the resource, in particular, interface description or other information important for the Incident investigation
Job/Session, **JobID/ SessionID** and **JobData/SessionData** – describing Job (or Session) submitted to the service or resource that has relation to the incident</td>
</tr>
</table>

	Authentication – contains information about the way the involved in the incident Principal was authenticated, including the case of failed authentication if this is classified as an incident **Authorization** - contains information about the way the involved in the incident Principal was authorized to access the Web Service or perform a specific action, including the case of failed authorization if this is classified as a incident **WebServiceData** – contains any additional information that describes the Web Service in details, including WSDl file or its components PortType, Binding, MessagePart; the element may also contain Request or Response messages that has relation to the incident

Source XML Schema	`<xs:complexType name="XMLWebServiceType">`
	`  <xs:sequence>`
	`    <xs:element ref="iodef-xws:EPReference" minOccurs="0"/>`
	`    <xs:element ref="iodef-xws:Principal" minOccurs="0" maxOccurs="unbounded"/>`
	`    <xs:element ref="iodef-xws:Resource" minOccurs="0" maxOccurs="unbounded"/>`
	`    <xs:element ref="iodef-xws:Job" minOccurs="0" maxOccurs="unbounded"/>`
	`    <xs:element ref="iodef-xws:Authentication" minOccurs="0" maxOccurs="unbounded"/>`
	`    <xs:element ref="iodef-xws:Authorization" minOccurs="0"/>`
	`    <xs:element ref="iodef-xws:WebServiceData" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Source XMLDTD	`<!ELEMENT XMLWebService (EPReference?, Principal*, Resource*, Job*, Authentication*, Authorization?, WebServiceData?)>`

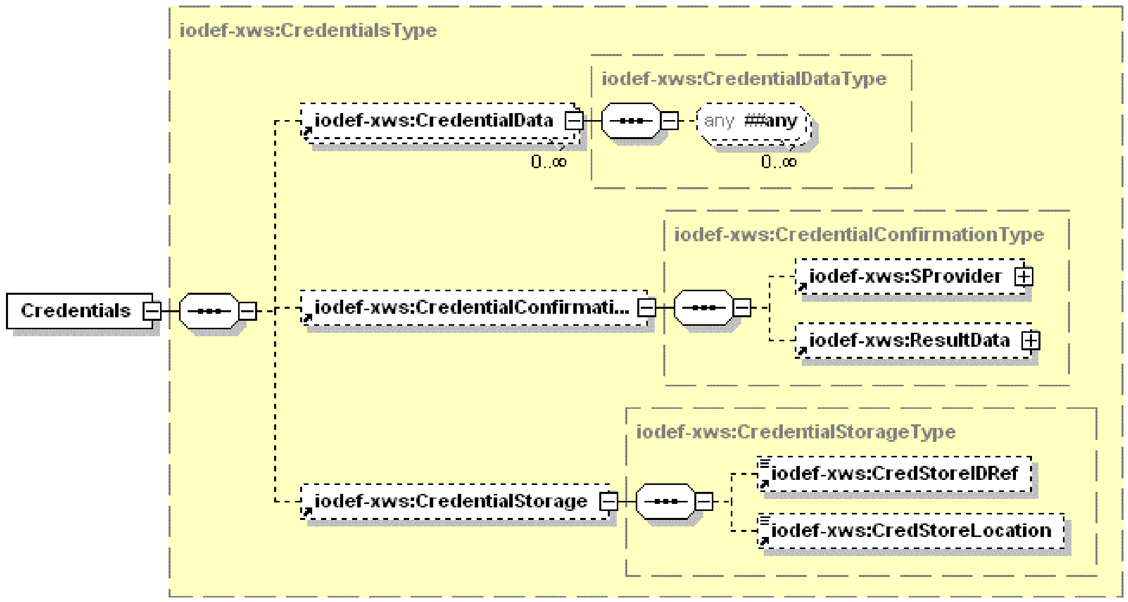
Element **Principal** contains information about the entities participating in the Web/Grid Services interaction. It may be a user or other entity who owns the service, or on behalf of which the action or service were requested that were involved in the incident.

diagram	 iodef-xws:PrincipalType iodef:NameIdentifier iodef-xws:CredentialsType iodef-xws:CredentialData 0..∞ Principal iodef-xws:Credentials iodef-xws:CredentialConfirmati... 0..∞ iodef-xws:CredentialStorage iodef-xws:Attributes
children	**NameIdentifier** – contains name that uniquely identifies the Principal; the element suggests different formats including but not limited to email address, X.509NameQualifier, URN or other used locally. **Credentials** – contains the credentials identifying the Principal and confirming the name contained in the NameIdentifier element; the Credentials element contains the sub-elements CredentialData, CredentialConfirmation and CredentialStorage **Attributes** – contains the attributes that are associated with the subject and are important for the incident investigation; attributes that include Principal's roles, group or association, and also possible restrictions

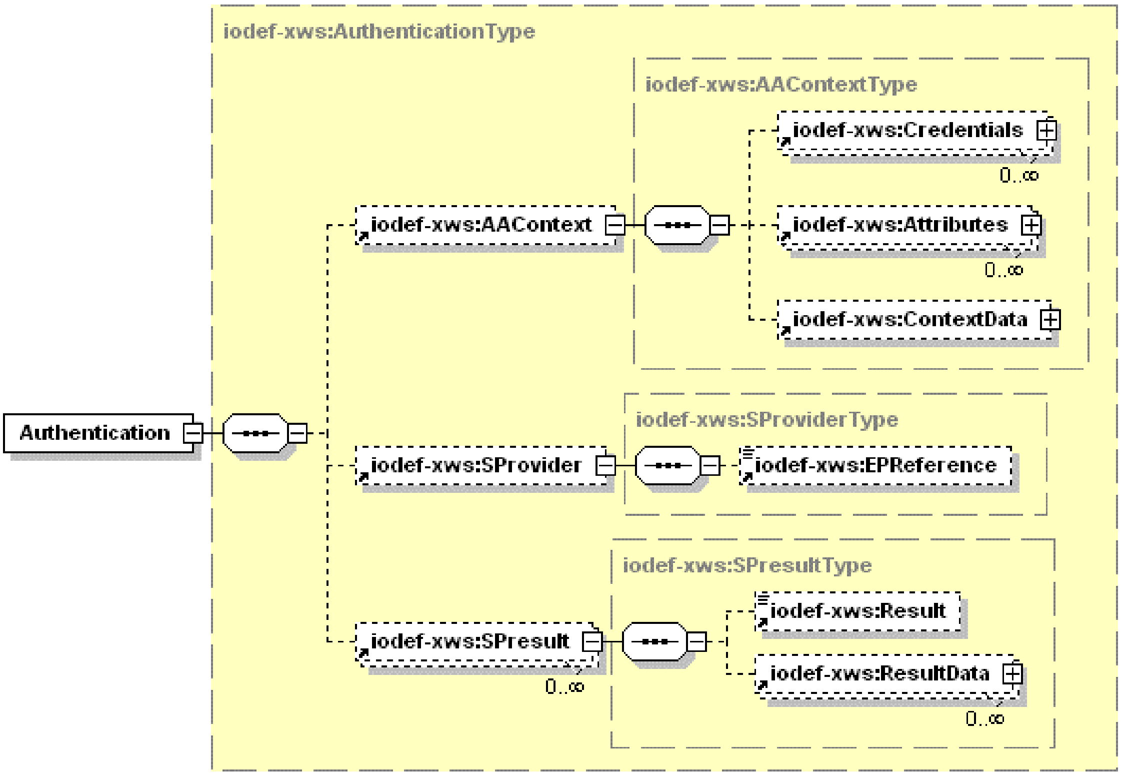
Source XML Schema	`<xs:complexType name="PrincipalType">`
	`  <xs:sequence>`
	`    <xs:element ref="iodef:NameIdentifier"/>`
	`    <xs:element ref="iodef-xws:Credentials" minOccurs="0" maxOccurs="unbounded"/>`

	<xs:element ref="iodef-xws:Attributes" minOccurs="0"/> </xs:sequence> <xs:attribute ref="iodef-xws:principalcat" default="other"/> </xs:complexType>
Source XML DTD	<!ELEMENT Principal (NameIdentifier, Credentials*, Attributes?)>

Element **Credentials** contains information about credentials related to principals involved in the incident.

diagram	 iodef-xws:CredentialsType iodef-xws:CredentialDataType iodef-xws:CredentialData any ##any 0..∞ 0..∞ iodef-xws:CredentialConfirmationType iodef-xws:SProvider Credentials iodef-xws:CredentialConfirmati... iodef-xws:ResultData iodef-xws:CredentialStorageType iodef-xws:CredStoreIDRef iodef-xws:CredentialStorage iodef-xws:CredStoreLocation
childrens	**CredentialData** – contains credential data in arbitrary format including X.509 Public Key Certificate (PKC), Attribute Certificate (AC), Proxy Certificate (Proxy), or XML assertions. **CredentialConfirmation** – may contain information about the Authentication Service Provider and/or about the result **CredentialStorage** – contains information about used credential storage including storage identification CredStoreIDRef and storage location CredStoreLocation Mandatory attribute of both Credentials and CredentialStorage elements is the storage status attribute ***"status"*** that indicates whether the credentials or storage are valid, protected, compromised, or quarantined.
Source XML Schema	<xs:complexType name="CredentialsType"> <xs:sequence> <xs:element ref="iodef-xws:CredentialData" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="iodef-xws:CredentialConfirmation" minOccurs="0"/> <xs:element ref="iodef-xws:CredentialStorage" minOccurs="0"/> </xs:sequence> <xs:attribute ref="iodef:restriction" default="default"/> <xs:attribute ref="iodef-xws:credstatus" use="required"/> </xs:complexType>
Source XML DTD	<!ELEMENT Credentials (uid?, Name?, Certificate+, AdditionalData*)>

Elements **Authentication and Authorization** contain information about AuthN/Z process and related data for entities or principals involved in the incident. These elements are also intended to provide a format for describing such security events as failed authentication or misused privileges.

diagram	 Authentication iodef-xws:AuthenticationType iodef-xws:AAContext iodef-xws:AAContextType iodef-xws:Credentials 0..∞ iodef-xws:Attributes 0..∞ iodef-xws:ContextData iodef-xws:SProvider iodef-xws:SProviderType iodef-xws:EPReference iodef-xws:SPresult 0..∞ iodef-xws:SPresultType iodef-xws:Result iodef-xws:ResultData 0..∞
childrens	**AAContext** – the element containing context of the authentication and authorisation decision and may include provided credentials, attributes or other arbitrary data **ContextData** – contains any arbitrary data that provide context for the authorisation or authentication decision **SProvider** – contains information about the authorisation or authentication service providers that in particular provided access control service for the principal **SPresult** – contains information about the result of principal's authorisation or authentication including the case of failed authorization if its was classified as an incident; SPresult contains the sub-elements Result and ResultData **Result** – describes the result of the principal's authentication and authorisation by the SProvider in a free from or from the restricted enumerated list **ResultData** – contains arbitrary data providing necessary background/context information supplementary to the result

Source XML Schema	`<xs:complexType name="AuthenticationType">`
	`  <xs:sequence>`
	`    <xs:element ref="iodef-xws:AAContext" minOccurs="0"/>`
	`    <xs:element ref="iodef-xws:SProvider" minOccurs="0"/>`
	`    <xs:element ref="iodef-xws:SPresult" minOccurs="0"`
	`maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`  <xs:attribute ref="iodef-xws:authnmethod" use="required"/>`
	`</xs:complexType>`

Source XML DTD	`<!ELEMENT Authentication (AAContext?, SProvider?, SPresult*)>`

XML based nature of interactions in Web Services and Grids provides good integration basis between Incident reporting facilities and applications and their monitoring tools. Thus, original data in XML format can be placed into the extensibility ***Data** elements that can adopt data formats from different namespaces than basic "iodef-xws" namespace.

The proposed extensions for description of the Grid Security Incident were discussed on IETF-INCH mailing list <inch@NIC.SURFNET.NL> [11] and presented to the INCH WG for consideration.

Modelling of proposed extensions with the XML schema is presented at the IODEF Schema information site http://www.uazone.org/demch/projects/iodef/ as IODEF version 0.42 [16].

9. Summary

Proposed analysis of the Web services and Grid security threats and risks provides an initial base for developers and practitioners for closer look at potential threats and vulnerabilities.

Web Services technologies and further their development in Computer grids open new kind of Security attacks and Incidents that can be defined as "white collar" attacks. Specifics of this kind of attacks from the point of view of applications and network protection is that malifactor is interested in correct and smooth work of a target system or application. Classically, white collar crime or commercial crime involves crimes such as fraud, ordinary theft, 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 misuse of WSDL information and tampering SOAP messages and 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 or the electronic identity in Grids and Web Services character of threats and security incidents will inevitably change with time.

The document presents also updates and further development of Grid Security Incident datamodel and XML schema as extension to the IODEF incident description format that is used by CSIRTs for incident information exchange. Requirement for general Incident reporting format FINE adopted by the CSIRT community is reviewed for its possible use for Grid Security Incident reporting. The IODEF profile for XML Web Services and Grids is proposed based on general XML Web Services and Grid vulnerabilities analysis and basic services operational security model. It is perceived that XML based nature of interactions in Web Services and Grids can provide even better integration between Incident reporting facilities and applications and their monitoring tools than in network applications.
The information on IODEF profile for XML Web Services and Grid have been presented to the CSIRT community for discussion.

References

LCG Risk Analysis – http://proj-lcg-security.web.cern.ch/proj-lcg-security/RiskAnalysis/risk.html

Incident Response General Issues, by Demchenko, Yuri - 2nd Middleware Security meeting, June 16, 2004. - http://agenda.cern.ch/askArchive.php?base=agenda&categ=a042157&id=a042157s15/transparencies

Anatomy of a Web Services Attack: A Guide to Threats and Preventative Countermeasures - Forum Systems, Inc., http:// www.forumsystems.com - March 1, 2004 - http://whitepapers.itsj.com/detail/RES/1084293354_294.html

Attacking and Defending Web Services. A Spire Research Report. – January 2004 Spire Security, LLC, http://www.spiresecurity.com - January 1, 2004 - http://whitepapers.itsj.com/detail/RES/1075225294_11.html

A Guide to Securing XML and Web Services. - ZapThink, LLC - January 1, 2004 - http://whitepapers.itsj.com/detail/RES/1073404572_221.html

Dane Skow "A walk through a Grid Security Incident" - http://www.triumf.ca/hepix2003/pres/23-10/dskow/A%20walk%20through%20a%20Grid%20Security%20Incident-v2.ppt

TeraGrid User News: Security Incident Notice. - April 2004 - http://news.teragrid.org/announcements/archive/20040407_02.php

Summary of UF/IU security incident – June 2004 - http://www-mcs.ivdgl.org/mail_archive/grid3-all/2004/06/msg00048.html

IETF INCH WG - http://www.ietf.org/html.charters/inch-charter.html

Unofficial IODEF website - http://www.cert.org/ietf/inch/inch.html

IODEF Schema information site - http://www.uazone.org/demch/projects/iodef/

Appendix A. Grid security threats and risks and required IODEF description elements

ID	Description	Evidence (what, where) and required IODEF elements
Technology platform (XML Web Services) vulnerabilities and threats		
T1	WSDL Scanning	
T2	WSDL Parameter Tampering	
T3	Recursive XML document (payload) content	
T4	Oversized XML documents/payloads	
T5	Malicious code exploiting known vulnerabilities in applications	
T6	Viruses, or Trojan horse programs	
T7	Malicious XPath or XQuery built-in operations	
T8	SQL Injection	
T9	Malicious XML Schema extensions (so called Schema Poisoning)	
T10	External Entity Attack	
T11	SOAP Flooding Attack (DoS)	
T12	Replay Attacks	
T13	Routing Detours	
T14	Message eavesdropping	
T15	"Man-in-the-middle" attack	
Confidentiality and Data integrity issues		
C1	Theft of credentials, e.g. private keys	File access, Record/Log (**patterns**), **Credentials**, Impact, **Principal/Identity (Victim)**
C2	Data or passwords/pass phrases exposed, e.g. in unprotected files or on the network	Yet Not Incident – Just risk
C3	Falsification of scientific data, analysis and/or results	File access, Log, Record/**Data**
C4	Unauthorized monitoring of network communications	System/process, Record (Registry)
C5	Unauthorized access to data	Log, **Data** or FileList
C6	Unauthorized distribution or exposure of data	**Data/uri** or File, log
C8	Identity or usage information is harvested by unauthorized persons	System/process, Record (Registry)
C9	Security assertions (AuthN or AuthZ token, etc.) tampering or hijacking	
Disruption of LCG infrastructure for political or other reasons		
D1	Disruption via exploitation of security holes	Ordinary attack (System, Contact, Method, Record)
D2	Corruption of or damage to data	**Data/uri** or File, log
D3	DOS attacks towards LCG to prevent normal working of network or services	Ordinary attack (multiple System, Contact, Method, Record)
D5	"Poisoned" resources are deployed on LCG to confuse operations, debugging or results	**Data/uri** or File, log, source system – the same as Data modification

D6	Attack by disgruntled users, employees or ex-employees	Ordinary attack (multiple System, Contact, Method, Record)
D7	Use of "social engineering" methods to attack LCG resources	Mostly resulted in theft of credentials
D8	Damage caused by viruses, worms, trojans or back-doors	Level of ordinary attacks
D9	Misleading trouble reports to the GOC or incident response mechanisms, to disrupt operations or damage reputation	Related to Incident Handling System – should be secured by mutual AuthN
D10	Modification or defacement of User Interfaces, documentation, monitoring etc, for disruption or advertising	Ordinary attack
Misuse of LCG resources - CPU, storage, network etc		
M1	Resources used to launch online attacks on other sites via DOS, Virus, Worms, SPAM etc	System, Contact, Record/logfile
M2	Resources used for offline attacks on other sites, e.g. to crack passwords or pass phrases	**Data**, Record/log/**pattern**
M3	Resources used to distribute or share non-LCG data, e.g. copyrighted, illegal, or inappropriate material	**Data/uri**, addData (sys image)
M4	Resources misused by inappropriate setting of access control or priority	System modification, FileList, User?/
M5	Use of LCG resources by unauthorized parties	Log, **Credentials (Attacker)**
M6	Use of LCG resources for unauthorized purposes, e.g. financial gain	Log, **Credentials (Attacker)**, impact
Security Issues - Non-intentional or accidental		
A1	Unauthorized use resulting from insecure middleware or bad security design/implementation	Risk – not incident
A2	Development process results in insecure middleware	Risk – not incident
A3	Deployment process results in insecure middleware	Risk – not incident
A4	Development process results in poor fault tolerance and effective loss of service (snowballing failure)	Risk – not incident
A5	Deployment process results in poor fault tolerance and effective loss of service (snowballing failure)	Risk – not incident
A6	Failure to perform security audit of new software	Risk – not incident
A7	Lack of timely patching of systems and middleware for security holes	Risk – not incident
A8	The need to incorporate legacy resources/applications prevents addressing security holes	Risk – not incident
A9	Problems from misleading or missing documentation	Risk – not incident
A10	Lack of critical security services, e.g. CRL's at CA's	
A11	Hardware faults	May lead to an Incident
A12	Disasters, e.g. fire or flood	Risk – not incident
A13	Accidental corruption of or damage to data	May be investigated as an Incident – ordinary Data/File corruption/modification
A14	Lack of knowledge and/or insufficient training of management, operations and support staff	Risk – not incident
A15	Security Infrastructure is not well matched to user requirements or expectations, and therefore too restrictive or too open	Risk – not incident
A16	LCG Authorization controls are insufficient to allow effective management by VO's, groups or users	Risk – not incident. Should not happen at all, Sec/AuthZ service must

		use existing tools.
Other attacks		
O1	Theft of systems	Not technical
O2	Theft of software	-“-
O3	Physical sabotage of systems	-“-
O4	Theft of primary or backup data media	-“-