Oval 5.x Services Orientated Architecture: Interpreter Services Proposal

[OVAL:SOA:IS]

Ken Lassesen, Patchlink.com
Loren Bandiera, MMG Security

Intellectual Property Statement
PatchLink/MMG Security grants the OVAL™ community an unrestricted use license for any content of this document when incorporated into OVAL™’s official schema and official standards.
1. Table of Contents

1. Table of Contents .................................................................................................................. 2
   1.1 RSA Expo OVAL Demonstration 2007 ............................................................................. 2
2. Objectives ............................................................................................................................. 3
3. Common SOA Features ......................................................................................................... 5
   3.1 Compression (GetCompressionsSupported) ................................................................... 5
   3.2 Security (GetSignatureXml) ........................................................................................... 6
   3.3 Return Values (GetError) ............................................................................................... 6
   3.4 Notation ......................................................................................................................... 7
   3.5 Load Balancing (ServerRequestedWait) ...................................................................... 7
   3.6 Notation ......................................................................................................................... 7
4. Common Services ................................................................................................................ 8
   4.1 GetSignatureXml ............................................................................................................ 8
   4.2 GetCompressionsSupported ......................................................................................... 8
   4.3 GetError ....................................................................................................................... 8
   4.4 ServerRequestedWait ................................................................................................. 9
5. Interpreter Web Services .................................................................................................. 10
   5.1 RequestClientID .......................................................................................................... 13
   5.2 GetOvalSchemaVersion ............................................................................................. 13
   5.3 GetDefinitions ............................................................................................................ 13
   5.4 ReturnResults ............................................................................................................. 15
   5.5 ReturnSystemCharacteristics .................................................................................... 15
6. Extended Interpreter Web Services ................................................................................. 16
   6.1 RequestXmlSignature ................................................................................................. 16
   6.2 GetSchema .................................................................................................................. 16
   6.3 GetSchedule ............................................................................................................... 17
7. Implementations ................................................................................................................ 19
   7.1 Sussen .......................................................................................................................... 19
   7.2 POIW ........................................................................................................................... 19
   7.3 Reference Web Server ............................................................................................... 19
   7.4 Latest Version of this document ............................................................................... 19
8. WSDL: Service Description .............................................................................................. 20
9. Example of an application using this web service .............................................................. 31
10. Revision History ............................................................................................................... 33

1.1 RSA Expo OVAL Demonstration 2007

PatchLink and MMGSecurity have implemented the interface described in this document and plan to demonstrate it (with any other implementers who conform to this interface) at this conference. If you wish to have access to the reference implementation, please email Ken.Lassesen@PatchLink.com or lorenb@mmgsecurity.com
2. Objectives

PatchLink proposes a series of Service Orientated Architecture [SOA] implementations for inclusion into OVAL standards. The Services Orientated Architecture model is growing in popularity and has a host of benefits, including

- Longer life-span of components and systems
- Simpler system
- Lower costs of implementation
- In philosophical alignment with the standards movement.

Purposes of these SOA proposals include:

- Encourage co-operative development and interoperability between vendor products.
- Encourage easy cross validation of results from different vendor products to improve the quality of all products.
  - Improve the ability for Mitre to be able to certify systems in more aspects.
- Encourage “best of breed” solutions by allowing users to mix and match due to an open plug-and-play architecture.
- Encourage “nitch” vendors to excel in their expertise instead of being force to invest heavily in a broad solution across the entire solution space.
  - Reduce the cost of a nitch vendor to enter the market
  - Increase the marketing opportunity for nitch vendors

Example:
A nitch vendor who has great expertise with various Apple OS’s may develop an excellent OVAL Interpreter. If this interpreter conforms to the SOA Client implementation then this vendor does not need to produce a complete OVAL system, instead, they can sell their interpreter to customers directly and / or to other vendors for inclusion in their packages. The larger vendors receive the benefit of reduced capital costs and a component that is likely superior to what they could have produced.

This document arose from a partnering with MMG Security and addresses the Interpreter Implementation [SOA Interpreter], that is the communications between between a host based client [CLIENT] and a data distribution and collection server [SERVER].. Other implementations in this SOA include:

- SOA Data Service – aggregation and detail interface allowing GUI systems to display data [OVAL:SOA:DS]
- SOA Remedy Service – an interface that allows remedies (such as those described in PatchLink Remedy Proposals) to be implemented at clients. [OVAL:SOA:RS]
- SOA Consolidation Service – an interface to a consolidating repository. Allows new definitions to be retrieved, problems reported and updated definitions delivered.[OVAL:SOA:PS]

The authors’ personal hope is that this approach would shift the participants in the community in a co-operative direction instead of competitive with *winner take all*. The author believes that dominance of a single vendor in this area will increase the security risk to the national’s agencies and corporations.
Ken.Lassesen@patchlink.com or lorenb@mmgsecurity.com
3. Common SOA Features

The following are methods desirable to have in each service offering.

Figure 1 Common WebServices

- **GetCompressionsSupported**
  Returns the forms of compression supported.

- **GetError**
  Returns a textual message for an error code/These will vary between vendor except that 0 means success.

- **GetSignatureXml**
  Returns the public XML Signature used to sign definitions.xml.

- **ServerRequestedWait**
  Returns the number of seconds that the server is requesting the client to wait inorder to do load balancing, etc.

All of these may be implemented with only a stub/constant response (shown below). If the service does not support the method, then the value show should be assumed.

- GetCompressionsSupported. Return “none”
- GetError: “No Information Available”
- GetSignatureXml: null
- ServerRequestedWait: Zero (0) Seconds – no wait.

3.1 Compression (GetCompressionsSupported)

Data compression is a desired characteristic for all SOA that allows good performance on low bandwidth connections. For illustration, compression with ZIP was done with the results shown

<table>
<thead>
<tr>
<th>File</th>
<th>Percentage Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition.xml</td>
<td>91%</td>
</tr>
<tr>
<td>Results.xml</td>
<td>93%</td>
</tr>
<tr>
<td>System-Characteristic.xml</td>
<td>93%</td>
</tr>
</tbody>
</table>

The following compression types are recommended to be supported as a minimum set:

- “tar” – typically for classic UNIX
- “zip” – typically for Windows
- “bzip2” – typically for RedHat
- “tgz” – a tar with gzip, a.k.a. “tarball”
- “gzip” -

In the APIs below this is represented by the parameter name “compressionType”. Compression applies to all data with a byte[] data type.

If a compression is specified, all byte[] sent to the server must be compressed using the specified compression, which will also be the compression any byte[] will be return in
3.2 Security (GetSignatureXml)

It is recommended that all critical files include xml signatures. It is suggested that the physical name of the public key file follow the reverse domain naming practice of the web site that the service is on.

Example:

The advantage of keys over a separate MD5 value is that once the initial communications has been established, there is never a need to re-request the key. With a MD5 there is a need to request it on every file. Such requests are a security vulnerability because both the definitions file and the MD5 can be intercepted and replaced. Additionally, because http requests are stateless both the MD5 and the data must be returned in the same request. A signature file may be delivered through https:, included in the installation package or by hand to eliminate the risk of intercept and replace. This approach allows plain http to be used for transmitting definitions. There is no need to encrypt the definitions (which can be counter productive for compression).

3.3 Return Values (GetError)

Most calls return an integer value. These values may vary from vendor to vendor according to their implementations. The values between 1000 and -1000 are reserved to the specification. A negative number indicates a failure, a positive number indicates a warning (i.e. the data did not validate against the schema but there was some data that could be processed successfully).
### Table 1 Reserved Return Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-8</td>
<td>Method does not support the OVAL Schema version specified</td>
</tr>
<tr>
<td>-7</td>
<td>Compression format requested does not support entries. Entry support is required for this method.</td>
</tr>
<tr>
<td>-6</td>
<td>Data does not contain OVAL Schema version</td>
</tr>
<tr>
<td>-5</td>
<td>File Permissions Problems on Server</td>
</tr>
<tr>
<td>-4</td>
<td>Unexpected fatal error</td>
</tr>
<tr>
<td>-3</td>
<td>Decompression failed</td>
</tr>
<tr>
<td>-2</td>
<td>Failed to match schema – no processing occurred</td>
</tr>
<tr>
<td>-1</td>
<td>Not valid XML</td>
</tr>
<tr>
<td>0</td>
<td>Success</td>
</tr>
<tr>
<td>1</td>
<td>Function is stubbed at the moment. Assume success</td>
</tr>
<tr>
<td>2</td>
<td>Failed to match schema – processing occurred and data was found</td>
</tr>
<tr>
<td>3</td>
<td>Success – but no data was changed. Data may have already been sent or is stale.</td>
</tr>
</tbody>
</table>

A verbose description of any errors may be obtained from `GetError`.

### 3.4 Notation

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Indicates a concept that is stored as a node</td>
</tr>
<tr>
<td>@name</td>
<td>Indicates a concept that is stored as an attribute</td>
</tr>
<tr>
<td>WebMethod</td>
<td>Indicates a web method call.</td>
</tr>
</tbody>
</table>

- The plain English meaning of the error code may be obtained.

### 3.5 Load Balancing (ServerRequestedWait)

Servers do not have unlimited resources and when available resources are exceeded may hang, timeout or crash. To prevent this, the `ServerRequestedWait` may be used to tell service clients to go away for a while and then come back.

### 3.6 Notation

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Indicates a concept that is stored as a node</td>
</tr>
<tr>
<td>@name</td>
<td>Indicates a concept that is stored as an attribute</td>
</tr>
</tbody>
</table>
4. Common Services

4.1 GetSignatureXml
This API returns the public (read) xml signature file for the web service/site. The vendor may elect to return an empty string if they do not intend to make the signature openly available (i.e. the signature may be available only by subscription).

```java
public int GetSignatureXml(
    string compressionType,
    out byte[] data
)
```

Example:
```xml
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://oval.mitre.org/">  
<RSAKeyValue>
  <Modulus>02Lf2sIDpzpxxb9NhDYKGJEZffmAaTfu5IoN9khlQysfUM2oeRJO/PsM6j6Yae0EuGZ+Hm6L4mDBIt9J0B
  R3PHvss9YiEcovchb2JOxTm9wHCMyWBM/WiuVFWx0y+f25AIguqAdt25C4K
  QxGXr9paWm1UC454m18fdFlU/</Modulus>
  <Exponent>AQAB</Exponent>
  <P>7KiX+p/3M1qUkwZkGwpYzPXM54idyk1vVB0FLvHW5b5JaDwAZcn1MKq9
  fjg2h13eDiAeUlsYJSVbvQ==</P><Q>5KmHcD1nRipmHlkKnVzl1J+egT9dyf4+
  6K22y0UoYcut+m6IKb/qXD6fIw6v0vHB94VLTUTyiI2My2Zb5BfeQ==</Q><DP>
  aXYsmp3Uo780cQ14KtxK+ItdAryqdfwXfwFTnRIS+TW0fbYs6BcfJ0B9Q1AtIjaPPk
  Da5du/a306n5GeB7+5Q==</DP><DQ>fGxXG1b+NCrSnyArxsfqSVdVv4Smd1Z
  WSuwDFy2C4KId4DPR1+cb=VPKBiZxnmRw13mTPCEJl/VQEynFGQ==</DQ>
</RSAKeyValue>
</string>
```

4.2 GetCompressionSupported
This API returns a string containing the type of compression supported by the service. It is a semi-colon delimited list.

```java
public string GetCompressionSupported()
```

Example of response:
```xml
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://oval.mitre.org/">none,bzip2,zip,gzip,tar</string>
```

4.3 GetError
Returns a textual message for an error code. These will vary between vendor except that 0 means success.

```java
public string GetError(
    int errorcode
)
```
Example:
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://oval.mitre.org/">Success</string>

4.4 ServerRequestedWait

Returns the number of seconds that the server is requesting the client to wait inorder to do load balancing, etc

    public int ServerRequestedWait()

This allows a server to implement some form of load balancing by allowing it to request clients to not submit load immediately.
A value of zero or less means that the client may make requests or submit data immediately.

Example:
<?xml version="1.0" encoding="utf-8" ?>
<int xmlns="http://oval.mitre.org/">0</int>
5. Interpreter Web Services

The Interpreter web services deals with passing information to and from the OVAL Interpreter [OI] installed on the client PC. The client-side consumer of these services may be built into the OI (internal consumer, for example Sussen\(^1\)) or may be a stand-alone component (external consumer, for example POIW\(^2\)) that invokes the OI, for example by spawning using command line arguments.

All OI with a built in consumer should continue to support command line arguments. It is recommended that a standard for command line arguments be also included in the OVAL specification to facilitate inter-operability.

**Figure 2 Example of Services produced by an IIS Server**

The following operations are supported. For a formal definition, please review the *Service Description*.

- **GetCompressionSupported**
  Returns the forms of compression supported.

- **GetDefinitions**
  Returns Xml Definition for the client. systemInfo must be sent with the same compression. The definitions should be digitally signed.

- **getError**
  Returns a textual message for an error code. These will vary between vendor except that 0 means success.

- **GetOvalSchemaVersion**
  Returns the OVAL Schema Version that the web service is using. The Schema files must be available for download.

- **GetSchedule**
  Returns the scheduling parameters (Time of day to run, hours between runs)

- **GetSchema**
  Get the Schemas(XSD) used by the current Schema Version

- **GetSignatureXml**
  Returns the public XML Signature used to sign definitions:xml

- **RequestClientID**
  Returns an identification node for this PC.

- **ReturnResults**
  Allows results to be returned to server from the client

- **ReturnSystemCharacteristics**
  Allows system characteristics to be returned to server from the client

- **ServerRequestedWait**
  Returns the number of seconds that the server is requesting the client to wait inorder to do load balancing, etc

The following diagram illustrates a potential implementation scenario with this approach, the names used do not indicate that the firms have this capacity but are simply used for illustrations (this applies to all diagrams in this document):

\(^1\) [http://dev.mmgsecurity.com/projects/sussen/](http://dev.mmgsecurity.com/projects/sussen/)

\(^2\) POIW – PatchLink OVAL Interpreter Wrapper
As you can see in the above diagram, it allows a diverse firm to be able to pick the best of breed for each environment they have. It also enables a smaller firm to remain competitive by not requiring them to produce components for every system. In short, it potentially reduces the cost of entry into OVAL by significantly reducing the overhead to do a marketable implementation (which could include 3rd party components). It also opens the market for smaller (or new) vendors to become component makers.
5.1 RequestClientID

This API allows a client to request a vendor specific id to uniquely identify it returns. The XML string return may be a <host_identification> node which should be included in the system_info sent to the browser.

The parameters sent is a single string (which may be empty) which may contain the <system_info> node.

```csharp
public int RequestClientID(
    string compressionType,
    byte[] systemInfo,
    out byte[] data
)
```

**Parameters**

- `system_info`: – expected to match <system_info>
  - May be empty
  - If it contains <host_identification> from this service, it should return it and *not* issue a new identifier.

Example:
```xml
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://oval.mitre.org/"> <host_identification url="LASSEPAD">a308cbd8-5b53-4f33-9801-65ab0a05637d</host_identification> </string>
```

5.2 GetOvalSchemaVersion

This API returns the OVAL Schema Version that the web service is using. If the client is not compatible with this version, then further communications should be terminated.

```csharp
public string GetOvalSchemaVersion()
```

Example of response:
```xml
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://oval.mitre.org/">5.1</string>
```

**Note:** that the server may require results and system-characteristics to be specified in this version. See Table 1 Reserved Return Values.

5.3 GetDefinitions

This API returns a string containing a <definitions> node (with child nodes) that is to be evaluated by this client. An external consumer would typically write this to the definitions.xml file.
The parameters sent is a single string which may be either the <system_info> node or an <affected> node (without children) which indicates the class of client. It is assumed that the <affected> node would be sent initially and <system_info> node will be subsequently sent.

**Comment:** This API can support the implementation of delta definitions (i.e. sending only new definitions, or definitions that need to be checked / confirmed). The use of deltas is a vendor choice.

```csharp
public int GetDefinitions(
    string compressionType,
    string prefix,
    byte[] systemInfo,
    out byte[] data
)
```

**Parameters**

- **prefix** – String – one of the values below:
  - “hpux”
  - “independent”
  - “linux”
  - “macos”
  - “solaris”
  - “unix”
  - “windows”

- **system_info** - matching <system_info>
  - May be empty
  - If it contains <host_identification> from this service, it should return it and not issue a new identifier.

- **data** - definitions.xml
5.4 ReturnResults
This API allows a client to return the results.xml file to the server.

```csharp
public int ReturnResults(
    string compressionType,
    byte[] systemInfo,
    byte[] data
)
```

The system_info parameter may be null, in that case the <system_info> in results should be used. A file "system_info.xml" is suggested to allow system configuration to change while maintaining identity and independence from the specific interpreter implementation.

Parameters
- data - oval_results – matching <oval_results>

5.5 ReturnSystemCharacteristics
This API allows a client to return the system-characteristics.xml file to the server.

The parameter is the contents of system-characteristics.xml are sent as a string.

```csharp
public int ReturnSystemCharacteristics(
    string compressionType,
    byte[] systemInfo,
    byte[] data
)
```

Parameters
- data - matching <oval_system_characteristics>
6. Extended Interpreter Web Services

The following web methods are extensions that may be available in some implementations. The purpose of these methods is to provide some standardization of possible client-server interactions.

6.1 RequestXmlSignature

This API returns a signing key for the client to sign their upload files. Each client should be assigned a different key. The signing key should be protected on the client through encryption, etc. The server should retain the reading key and delete the signing key.

```java
public int RequestSignatureXml(
    string compressionType,
    byte[] systemInfo,
    out byte[] data)
```

In a sensitive environment, the results should be signed to prevent men-in-the-middle attacks (for example, falsely reporting results so vulnerabilities can continue to be exploited on clients).

6.2 GetSchema

Returns in a format supporting multiple files(zip, tar) the collection of schema files associated with the current OVAL Schema Version.

```java
public int GetSchema(
    string compressionType,
    out byte[] data)
```

The files contained in the collection should not include paths. If a server implements extensions to OVAL, this allows those extensions' schemas to be downloaded. Some clients validates the definitions prior to evaluations, this allows the validation to occur even though a test may be unknown.
6.3 GetSchedule

Returns in XML the schedule for when this client is to execute. There appear to be no existing standard for schedules and the following simple format is proposed.

```java
public int GetSchedule(
    string compressionType,
    byte[] systemInfo,
    out int minHour,
    out int maxHour,
    out int intervalHour
)
```

Examples:
<table>
<thead>
<tr>
<th>minHour</th>
<th>maxHour</th>
<th>IntervalHour</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>24</td>
<td>6</td>
<td>Every 6 hours</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>18</td>
<td>Every 18 hours, between 6pm and 8am</td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>23</td>
<td>Every 23 hrs between 6pm and midnight</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>144</td>
<td>Every 7 days between 6pm and midnight</td>
</tr>
</tbody>
</table>
7. Implementations

7.1 Sussen
This is an OVAL interpreter that natively supports this web service.

Please get the latest source available at:
- Linux: http://dev.mmgsecurity.com/src/sussen/trunk

7.2 POIW
This is a Windows application that talks to a web service and shell out oval interpreters. It will automatically detect default installations of Mitre’s OVALDI and MMGSecurity Sussen and configure the client portion appropriately.

To get a user manual or the latest source available email: Ken.Lassesen@PatchLink.Com or Ken.Lassesen@gmail.com. There is a separate document describing POIW in this series of proposals.

7.3 Reference Web Server
PatchLink maintains a reference implementation available at:
http://206.63.165.69:1081/SOA/IS.asmx

Additional functions such as the ability to do delta between result files may be seen at:
http://206.63.165.69:1081/SOA/

7.4 Latest Version of this document
The latest version may be obtained from
- http://OVAL.patchlink.com (planned) or
8. WSDL: Service Description

```xml
<?xml version="1.0" encoding="utf-8"?>
<wSDL:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://oval.mitre.org/
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
targetNamespace="http://oval.mitre.org/
xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/"
xmlns:schema elementFormDefault="qualified"
targetNamespace="http://oval.mitre.org/">
  <wsdl:types>
    <s:schema targetNamespace="http://oval.mitre.org/"
      elementFormDefault="qualified">
      <s:complexType name="GetCompressionSupported">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1"
            name="GetCompressionSupportedResponse" type="s:base64Binary" />
          <s:element name="RequestSignatureXml">
            <s:complexType>
              <s:sequence>
                <s:element minOccurs="0" maxOccurs="1"
                  name="compressionType" type="s:string" />
                <s:element minOccurs="0" maxOccurs="1"
                  name="systemInfo" type="s:base64Binary" />
              </s:sequence>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
</wSDL:definitions>
```
<s:element minOccurs="1" maxOccurs="1" name="GetSignatureXmlResult" type="s:int" />
<s:element minOccurs="0" maxOccurs="1" name="data" type="s:base64Binary" />
</s:sequence>
</s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="RequestClientID">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="compressionType" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="systemInfo" type="s:base64Binary" />
</s:sequence>
</s:complexType>
</s:element>
<s:element minOccurs="1" maxOccurs="1" name="RequestClientIDResult" type="s:int" />
<s:element minOccurs="0" maxOccurs="1" name="data" type="s:base64Binary" />
</s:sequence>
</s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="ServerRequestedWait">
<s:complexType />
</s:element>
<s:element minOccurs="1" maxOccurs="1" name="ServerRequestedWaitResult" type="s:int" />
<s:element minOccurs="0" maxOccurs="1" name="data" type="s:base64Binary" />
</s:sequence>
</s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="GetDefinitions">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="compressionType" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="prefix" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="systemInfo" type="s:base64Binary" />
</s:sequence>
</s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="GetDefinitionsResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="GetDefinitionsResult" type="s:int" />
<s:element minOccurs="0" maxOccurs="1" name="data" type="s:base64Binary" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="ReturnResults">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="compressionType" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="systemInfo" type="s:base64Binary"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ReturnResultsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ReturnResultsResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ReturnSystemCharacteristics">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="compressionType" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="systemInfo" type="s:base64Binary"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ReturnSystemCharacteristicsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ReturnSystemCharacteristicsResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetOvalSchemaVersion">
  <s:complexType />
</s:element>

<s:element name="GetOvalSchemaVersionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetOvalSchemaVersionResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetSchema">
  <s:complexType />
</s:element>

<s:element name="GetSchemaResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetSchemaResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType>
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="GetSchemaResult" type="s:int"/>
    <s:element minOccurs="0" maxOccurs="1" name="data" type="s:base64Binary"/>
  </s:sequence>
</s:complexType>
</s:element>
<s:element name="GetError">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="errorcode" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetErrorResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetErrorResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetSchedule">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="compressionType" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="systemInfo" type="s:base64Binary"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetScheduleResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetScheduleResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType>
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="minHour" type="s:int"/>
    <s:element minOccurs="1" maxOccurs="1" name="maxHour" type="s:int"/>
    <s:element minOccurs="1" maxOccurs="1" name="intervalHour" type="s:int"/>
  </s:sequence>
</s:complexType>
</s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="GetCompressionSupportedSoapIn">
  <wsdl:part name="parameters" element="tns:GetCompressionSupported"/>
</wsdl:message>
<wsdl:message name="GetCompressionSupportedSoapOut">
  <wsdl:part name="parameters" element="tns:GetCompressionSupportedResponse"/>
</wsdl:message>
<wsdl:message name="RequestSignatureXmlSoapIn">
  <wsdl:part name="parameters" element="tns:RequestSignatureXml"/>
</wsdl:message>
<wsdl:message name="RequestSignatureXmlSoapOut">
  <wsdl:part name="parameters" element="tns:RequestSignatureXmlResponse"/>
</wsdl:message>

<wsdl:message name="GetSignatureXmlSoapIn">
  <wsdl:part name="parameters" element="tns:GetSignatureXml"/>
</wsdl:message>

<wsdl:message name="GetSignatureXmlSoapOut">
  <wsdl:part name="parameters" element="tns:GetSignatureXmlResponse"/>
</wsdl:message>

<wsdl:message name="RequestClientIDSoapIn">
  <wsdl:part name="parameters" element="tns:RequestClientID"/>
</wsdl:message>

<wsdl:message name="RequestClientIDSoapOut">
  <wsdl:part name="parameters" element="tns:RequestClientIDResponse"/>
</wsdl:message>

<wsdl:message name="ServerRequestedWaitSoapIn">
  <wsdl:part name="parameters" element="tns:ServerRequestedWait"/>
</wsdl:message>

<wsdl:message name="ServerRequestedWaitSoapOut">
  <wsdl:part name="parameters" element="tns:ServerRequestedWaitResponse"/>
</wsdl:message>

<wsdl:message name="GetDefinitionsSoapIn">
  <wsdl:part name="parameters" element="tns:GetDefinitions"/>
</wsdl:message>

<wsdl:message name="GetDefinitionsSoapOut">
  <wsdl:part name="parameters" element="tns:GetDefinitionsResponse"/>
</wsdl:message>

<wsdl:message name="ReturnResultsSoapIn">
  <wsdl:part name="parameters" element="tns:ReturnResults"/>
</wsdl:message>

<wsdl:message name="ReturnResultsSoapOut">
  <wsdl:part name="parameters" element="tns:ReturnResultsResponse"/>
</wsdl:message>

<wsdl:message name="ReturnSystemCharacteristicsSoapIn">
  <wsdl:part name="parameters" element="tns:ReturnSystemCharacteristics"/>
</wsdl:message>

<wsdl:message name="ReturnSystemCharacteristicsSoapOut">
  <wsdl:part name="parameters" element="tns:ReturnSystemCharacteristicsResponse"/>
</wsdl:message>

<wsdl:message name="GetOvalSchemaVersionSoapIn">
  <wsdl:part name="parameters" element="tns:GetOvalSchemaVersion"/>
</wsdl:message>

<wsdl:message name="GetOvalSchemaVersionSoapOut">
  <wsdl:part name="parameters" element="tns:GetOvalSchemaVersionResponse"/>
</wsdl:message>

<wsdl:message name="GetSchemaSoapIn">
  <wsdl:part name="parameters" element="tns:GetSchema"/>
</wsdl:message>

<wsdl:message name="GetSchemaSoapOut">
  <wsdl:part name="parameters" element="tns:GetSchemaResponse"/>
</wsdl:message>

<wsdl:message name="GetErrorSoapIn">
  <wsdl:part name="parameters" element="tns:GetError"/>
</wsdl:message>

<wsdl:message name="GetErrorSoapOut">
  <wsdl:part name="parameters" element="tns:GetErrorResponse"/>
</wsdl:message>

<wsdl:message name="GetScheduleSoapIn">
...
<wsdl:portType name="InterpreterServicesSoap">
  <wsdl:documentation>
    Returns Xml Definition for the client. systemInfo must be sent with the same compression. The definitions should be digitally signed.</wsdl:documentation>
  <wsdl:operation name="RequestClientID">
    <wsdl:input message="tns:RequestClientIDSoapIn"/>
    <wsdl:output message="tns:RequestClientIDSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetDefinitions">
    <wsdl:input message="tns:GetDefinitionsSoapIn"/>
    <wsdl:output message="tns:GetDefinitionsSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ReturnResults">
    <wsdl:input message="tns:returnResultsSoapIn"/>
    <wsdl:output message="tns:returnResultsSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetSignatureXml">
    <wsdl:input message="tns:GetSignatureXmlSoapIn"/>
    <wsdl:output message="tns:GetSignatureXmlSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="RequestSignatureXml">
    <wsdl:input message="tns:RequestSignatureXmlSoapIn"/>
    <wsdl:output message="tns:RequestSignatureXmlSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetCompressionSupported">
    <wsdl:input message="tns:GetCompressionSupportedSoapIn"/>
    <wsdl:output message="tns:GetCompressionSupportedSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ServerRequestedWait">
    <wsdl:input message="tns:ServerRequestedWaitSoapIn"/>
    <wsdl:output message="tns:ServerRequestedWaitSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetSchedule">
    <wsdl:input message="tns:GetScheduleSoapIn"/>
    <wsdl:output message="tns:GetScheduleSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ReturnSystemCharacteristics">
    <wsdl:input message="tns:returnSystemCharacteristicsSoapIn"/>
    <wsdl:output message="tns:returnSystemCharacteristicsSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleResponse">
    <wsdl:input message="tns:GetScheduleResponseSoapIn"/>
    <wsdl:output message="tns:GetScheduleResponseSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ServerRequestedWaitSoapOut">
    <wsdl:input message="tns:ServerRequestedWaitSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetSignatureXmlSoapOut">
    <wsdl:input message="tns:GetSignatureXmlSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="RequestSignatureXmlSoapOut">
    <wsdl:input message="tns:RequestSignatureXmlSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetCompressionSupportedSoapOut">
    <wsdl:input message="tns:GetCompressionSupportedSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleSoapOut">
    <wsdl:input message="tns:GetScheduleSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleSoapIn">
    <wsdl:input message="tns:GetScheduleSoapIn"/>
  </wsdl:operation>
  <wsdl:operation name="ReturnSystemCharacteristicsSoapOut">
    <wsdl:input message="tns:returnSystemCharacteristicsSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ReturnSystemCharacteristicsSoapIn">
    <wsdl:input message="tns:returnSystemCharacteristicsSoapIn"/>
  </wsdl:operation>
  <wsdl:operation name="GetSignatureXmlSoapIn">
    <wsdl:input message="tns:GetSignatureXmlSoapIn"/>
  </wsdl:operation>
  <wsdl:operation name="RequestSignatureXmlSoapIn">
    <wsdl:input message="tns:RequestSignatureXmlSoapIn"/>
  </wsdl:operation>
  <wsdl:operation name="GetCompressionSupportedSoapIn">
    <wsdl:input message="tns:GetCompressionSupportedSoapIn"/>
  </wsdl:operation>
  <wsdl:operation name="ServerRequestedWaitSoapIn">
    <wsdl:input message="tns:ServerRequestedWaitSoapIn"/>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleSoapIn">
    <wsdl:input message="tns:GetScheduleSoapIn"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:documentation>
  Allows results to be returned to server from the client</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns the number of seconds that the server is requesting the client to wait inorder to do load balancing, etc</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns an identification node for this PC.</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns the public XML Signature used to sign definitions.xml</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns a private XML Signature to sign results.xml or system-characteristics.xml.</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns the forms of compression supported.</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns Xml Definition for the</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns an identification node</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns the number of seconds that the server is requesting the client to wait inorder to do load balancing, etc.</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Returns Xml Definition for the client. systemInfo must be sent with the same compression. The definitions should be digitally signed.</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Allows results to be returned to server from the client</wsdl:documentation>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  Allows system characteristics to be returned to server from the client</wsdl:documentation>
<wsdl:input message="tns:ReturnSystemCharacteristicsSoapIn"/>
<wsdl:output message="tns:ReturnSystemCharacteristicsSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetOvalSchemaVersion">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
Returns the OVAL Schema Version that the web service is using. The Schema files must be available for download.
</wsdl:documentation>
<wsdl:input message="tns:GetOvalSchemaVersionSoapIn"/>
<wsdl:output message="tns:GetOvalSchemaVersionSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetSchedule">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
Returns the scheduling parameters (Time of day to run, hours between runs)
</wsdl:documentation>
<wsdl:input message="tns:GetScheduleSoapIn"/>
<wsdl:output message="tns:GetScheduleSoapOut"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="InterpreterServicesSoap" type="tns:InterpreterServicesSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="GetCompressionSupported">
<soap:operation soapAction="http://oval.mitre.org/GetCompressionSupported" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="RequestSignatureXml">
<soap:operation soapAction="http://oval.mitre.org/RequestSignatureXml" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSignatureXml">
<soap:operation soapAction="http://oval.mitre.org/GetSignatureXml" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetError">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
Returns a textual message for an error code/ These will vary between vendor except that 0 means success.
</wsdl:documentation>
<wsdl:input message="tns:GetErrorSoapIn"/>
<wsdl:output message="tns:GetErrorSoapOut"/>
</wsdl:operation>
</wsdl:binding>
</wsdl:binding>
</wsdl:service>
<wsdl:definitions name="InterpreterServices">
<wsdl:schema targetNamespace="http://www.oval.mitre.org/XMLSchema/oval-schema-5" elementFormDefault="qualified" attributeFormDefault="lowercase">
<wsdl:documentation>
</wsdl:documentation>
</wsdl:schema>
</wsdl:definitions>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="RequestClientID">
<soap:operation soapAction="http://oval.mitre.org/RequestClientID" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ServerRequestedWait">
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDefinitions">
<soap:operation soapAction="http://oval.mitre.org/GetDefinitions" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReturnResults">
<soap:operation soapAction="http://oval.mitre.org/ReturnResults" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReturnSystemCharacteristics">
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetOvalSchemaVersion">
</wsdl:operation>
<wsdl:operation name="GetSignatureXml">
  <soap12:operation soapAction="http://oval.mitre.org/GetSignatureXml"
      style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RequestClientID">
  <soap12:operation soapAction="http://oval.mitre.org/RequestClientID"
      style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ServerRequestedWait">
  <soap12:operation soapAction="http://oval.mitre.org/ServerRequestedWait"
      style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDefinitions">
  <soap12:operation soapAction="http://oval.mitre.org/GetDefinitions"
      style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReturnResults">
  <soap12:operation soapAction="http://oval.mitre.org/ReturnResults"
      style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReturnSystemCharacteristics">
  <soap12:operation soapAction="http://oval.mitre.org/ReturnSystemCharacteristics"
      style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetOvalSchemaVersion">
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetSchema">
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetError">
  <soap12:operation soapAction="http://oval.mitre.org/GetError" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetSchedule">
  <soap12:operation soapAction="http://oval.mitre.org/GetSchedule" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>

</wsdl:binding>
</wsdl:service>
</wsdl:definitions>
9. Example of an application using this web service

The following illustrates a simple application that allows delta’s between results to be generated.

This produces a result file that allows differences between interpreters to be examined.

```
<oval_delta>
    <!-- Items are those where all reports are not identical: differt value, or not reporting when other are reporting -->
    <reports>
        <element reportid="44">OVAL Definition Interpreter [5.0]  Schema:5.1
At:11/14/2006 11:06:25 PM</element>
```

Click on "Download" below to download the XML reported

<table>
<thead>
<tr>
<th>Action</th>
<th>Client Time</th>
<th>Upload At</th>
<th>Schema</th>
<th>Prod Name</th>
<th>Ver</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Download]</td>
<td>12/1/2006 7:02:26 PM</td>
<td>12/1/2006 7:05:01 PM</td>
<td>5.1</td>
<td>OVAL Definition Interpreter</td>
<td>5.1 Build: 14</td>
</tr>
<tr>
<td>[Download]</td>
<td>12/1/2006 7:05:38 PM</td>
<td>12/1/2006 7:06:13 PM</td>
<td>5.1</td>
<td>OVAL Definition Interpreter</td>
<td>5.1 Build: 14</td>
</tr>
<tr>
<td>[Download]</td>
<td>12/7/2006 3:35:40 PM</td>
<td>12/7/2006 3:36:20 PM</td>
<td>5.0</td>
<td>sussen</td>
<td>0.33</td>
</tr>
<tr>
<td>[Download]</td>
<td>12/7/2006 3:36:36 PM</td>
<td>12/7/2006 3:37:00 PM</td>
<td>5.0</td>
<td>sussen</td>
<td>0.33</td>
</tr>
</tbody>
</table>

This produces a result file that allows differences between interpreters to be examined.
<element reportid="70">OVAL Definition Interpreter [5.1 Build: 14]
Schema:5.1 At:12/1/2006 7:05:38 PM</element>
<element reportid="73">sussen [0.33] Schema:5.0 At:12/7/2006 3:35:40 PM</element>
<element reportid="74">sussen [0.33] Schema:5.0 At:12/7/2006 3:36:36 PM</element>
</reports>
<elements>
<element id="oval:org.mitre.oval:def:1570">
<definition xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="error" version="1" reportid="44" />
<definition xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="false" version="1" reportid="70" />
<definition xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="false" version="1" reportid="73" />
<definition xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="false" version="1" reportid="74" />
</definition>
<element id="oval:org.mitre.oval:tst:35">
<test xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="false" version="1" reportid="70" />
<test xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="true" version="1" reportid="73" />
</test>
<element id="oval:org.mitre.oval:tst:57">
<test xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="false" version="1" reportid="70" />
<test xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="true" version="1" reportid="73" />
</test>
<element id="oval:org.mitre.oval:tst:30">
<test xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="true" version="1" reportid="70" />
<test xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" result="false" version="1" reportid="73" />
</test>
</element>
10. Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2006-10-10</td>
<td>Ken Lassesen</td>
<td>• Initial Draft for public circulation</td>
</tr>
<tr>
<td>1.0.1</td>
<td>2006-11-01</td>
<td>Loren Bandiera</td>
<td>• Reconciliation of PatchLink and MMG Security approaches.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ken Lassesen</td>
<td>• Sussen Implementation added</td>
</tr>
<tr>
<td>1.0.2</td>
<td>2006-11-16</td>
<td>Loren Bandiera</td>
<td>• Extended for Schema Version Issues</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ken Lassesen</td>
<td>• Optional Calls added</td>
</tr>
<tr>
<td>1.0.3</td>
<td>2006-12-06</td>
<td>Ken Lassesen</td>
<td>• Re-organized and parameter consistency issues corrected.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Added example of application.</td>
</tr>
<tr>
<td>1.0.4</td>
<td>2006-12-20</td>
<td>Ken Lassesen</td>
<td>• Add “RequestSignatureXml” to improve security.</td>
</tr>
</tbody>
</table>

**Intellectual Property Caveat**

The contents of this document may include concepts, algorithms or methodologies that may be the subject of one or more patent applications.