

Oval 5.x Services Orientated Architecture: Interpreter Services Proposal

[OVAL:SOA:IS]

Ken Lassenen, 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 GetCompressionSupported.....	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 Intrepreter. 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 Intrepreter Implementation [SOA Intrepreter], 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

- **GetCompressionSupported**
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 in order 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

File	Percentage Reduction
Definition.xml	91%
Results.xml	93%
System-Characteristic.xml	93%

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:

- <http://oval.mitre.org/reference/webservice.asmx> --> org.mitre.oval.signature.xml
- <http://oval.lassesen.com/demo/webservice.asmx> --> com.lassesen.oval.signature.xml

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

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

A verbose description of any errors may be obtained from **GetError**

3.4 Notation

Notation	Description
<name>	Indicates a concept that is stored as a node
@name	Indicates a concept that is stored as an attribute
WebMethod	Indicates a web method call.

- 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

Notation	Description
<name>	Indicates a concept that is stored as a node
@name	Indicates a concept that is stored as an attribute

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).

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

Example:

```
<?xml version="1.0" encoding="utf-8" ?>  
<string  
  xmlns="http://oval.mitre.org/"><RSAKeyValue><Modulus>02Lf2sI DpzpXxib9Nh  
DYKGJEZZfmAaTfU5I oN9khlQysfUM2oeRJO/PsM6j6YAe0EuGZ+Hm6L4mDBI t9JOB  
R3PHuvsP9YIECovchb2JOxTml k9wHCRMMyBWM/WiuVFwY0y+f25Al guqADt25C4K  
QxGXr9paWml UC454ml 8fkDf1U= </Modulus><Exponent>AQAB</Exponent><P  
>7KiX+P/3SAIQI/3M1qUkwZkGwpYzPXM54idyk1vVB0FLvHW5bJaDwAZcn1MKzq  
9fjg2h13eDI aEuLsYJSVbvQ== </P><Q>5KmHcD1nRipmhIkKnVAZL1J+egT9dyf4+  
6K22y0UoYcut+m6IKb/qXD6fl iW6OvHB94VLTUTyil 2My2Zb5BfeQ== </Q><DP>a  
xYSxmpUo7S0cQI 4KTxK+I tdAryqdfwXfwFTnRl S+TW0fbyS6BcfJOB9Q1AtI jaPPkk  
Da5du/a306n5Ge7B+5Q== </DP><DQ>fGxXG1b+NCrsQNYArxsfqSVdVu4SMd1Z  
WSuwDIFYcZCKdl7DPRll+cBVPKBiZxmugnRw13mTPCEjlvOOEynfGQ== </DQ><I n  
verseQ>GySQNIImDUKAwQP/9YgzapTdOZPHGI3wajDmMOP82K06CiRq8Iqugm+S4  
2RhgjGxCl gAgdGp/9iWdnQHXL+m3MA== </I nverseQ><D>jKy6SLzg7YM4Mphz3  
mOrTQTI e8uwt9te/CDCPxhy7tw2Vk5Kp0ZjLIDQGjkUFqBJyF72tpNbwpmc7B1gNRB  
LqUZ0tOAR9M7ZAYYOnTlZv0l lszuE1JBW60NzD5l FUWWutFnYMv5FQ0oZHIFo513a  
jPLGm+aBXAZ7qRVINjUKQ8E= </D></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.

```
public string GetCompressionSupported()
```

Example of response:

```
<?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.

```
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 in order 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 Sussen¹) or may be a stand-alone component (external consumer, for example POIW²) 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

InterpreterServices

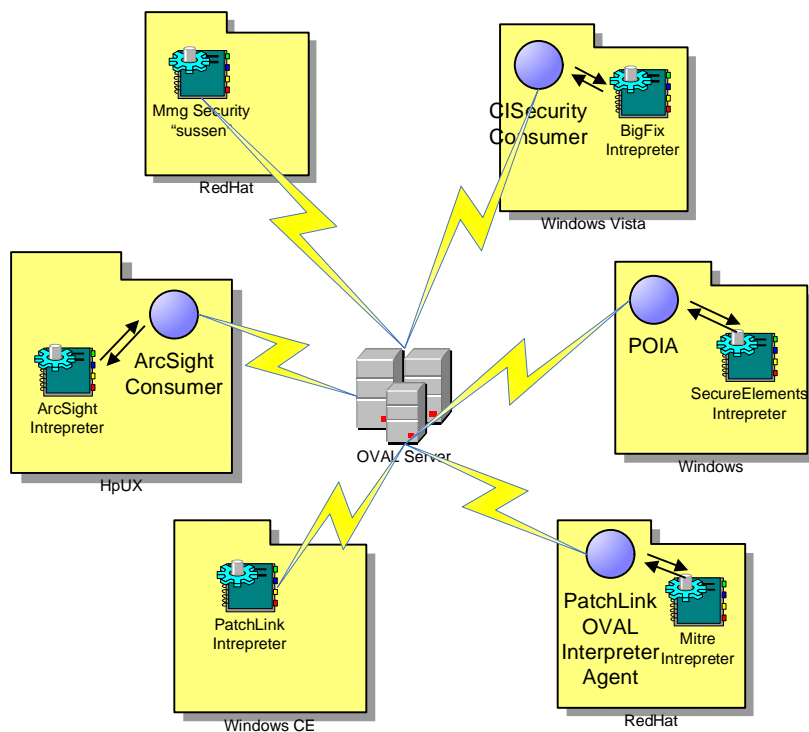
The following operations are supported. For a formal definition, please review the [Service Description](#).

- [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/>

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_indentifier> 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.

```
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 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.

```
public string GetOvalSchemaVersion()
```

Example of response:

```
<?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.

```
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.

```
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.

```
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.

```
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).

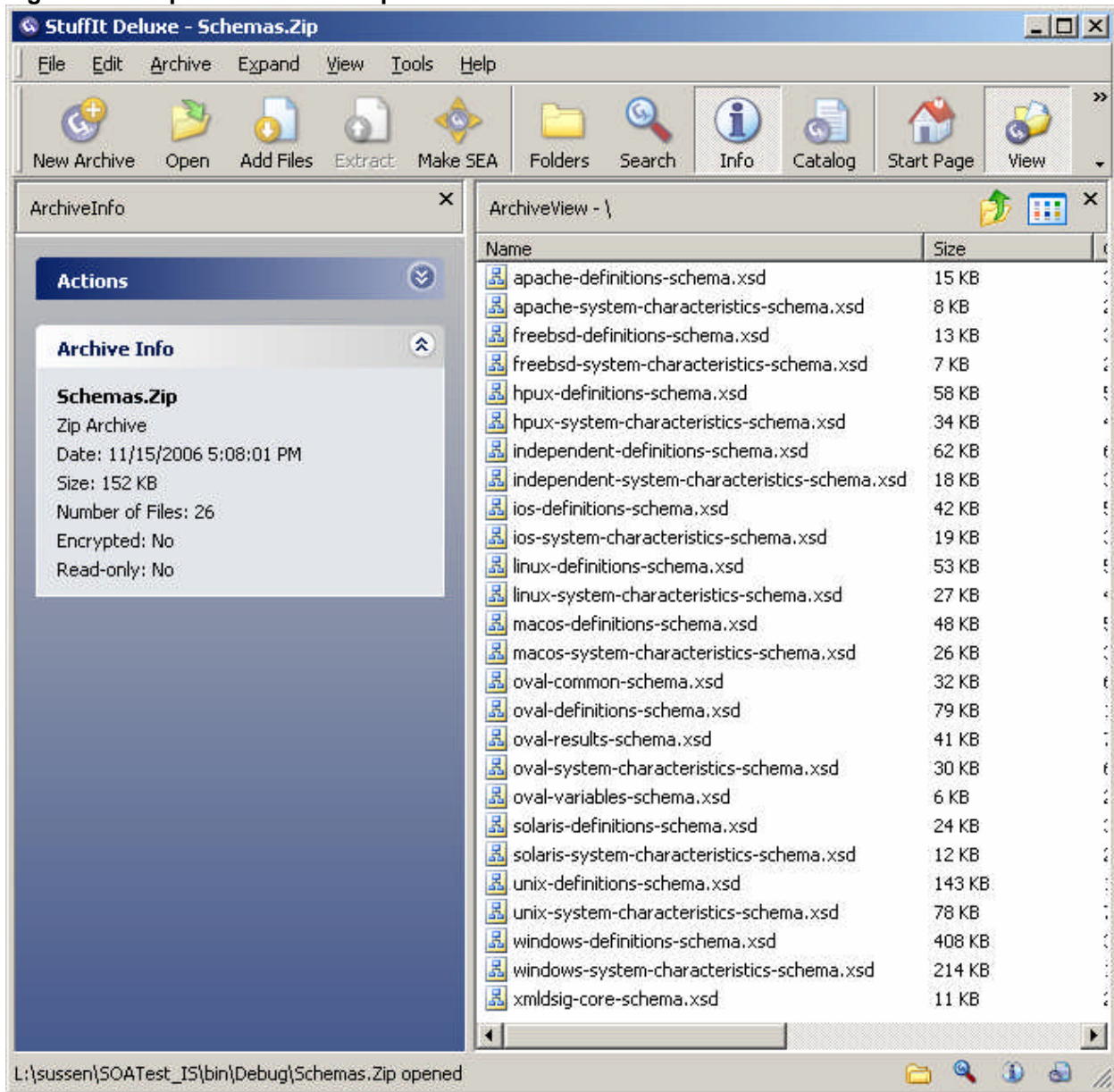
6.2 GetSchema

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

```
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.

Figure 3 Example of a Schema Zip file contents



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.

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

Examples:

minHour	maxHour	IntervalHour	Description
0	24	6	Every 6 hours
18	8	18	Every 18 hours, between 6pm and 8am
18	24	23	Every 23 hrs between 6pm and midnight
18	8	144	Every 7 days between 6pm and midnight

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>

Windows <http://dev.mmgsecurity.com/src/sussen/branches/sussen-win>

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
- <http://OVAL.Lassesen.com> (Temp site).

8. WSDL: Service Description

```
<?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/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://oval.mitre.org/">
      <s:element name="GetCompressionSupported">
        <s:complexType />
      </s:element>
      <s:element name="GetCompressionSupportedResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetCompressionSupportedResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <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:element name="RequestSignatureXmlResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1"
name="RequestSignatureXmlResult" type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSignatureXml">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSignatureXmlResponse">
        <s:complexType>
          <s:sequence>
```

```

        <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 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 name="RequestClientIDResponse">
    <s:complexType>
        <s:sequence>
            <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 name="ServerRequestedWait">
    <s:complexType />
</s:element>
<s:element name="ServerRequestedWaitResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1"
name="ServerRequestedWaitResult" type="s:int" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element 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 name="GetDefinitionsResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" 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:element minOccurs="0" maxOccurs="1" name="data"
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:element minOccurs="0" maxOccurs="1" name="data"
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:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetSchemaResponse">

```

```

        <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: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:part name="parameters" element="tns:GetSchedule" />
</wsdl:message>
<wsdl:message name="GetScheduleSoapOut">
  <wsdl:part name="parameters" element="tns:GetScheduleResponse" />
</wsdl:message>
<wsdl:portType name="InterpreterServicesSoap">
  <wsdl:operation name="GetCompressionSupported">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the forms of compression
supported.</wsdl:documentation>
  <wsdl:input message="tns:GetCompressionSupportedSoapIn" />
  <wsdl:output message="tns:GetCompressionSupportedSoapOut" />
</wsdl:operation>
  <wsdl:operation name="RequestSignatureXml">
    <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:input message="tns:RequestSignatureXmlSoapIn" />
  <wsdl:output message="tns:RequestSignatureXmlSoapOut" />
</wsdl:operation>
  <wsdl:operation name="GetSignatureXml">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the public XML Signature
used to sign definitions.xml.</wsdl:documentation>
  <wsdl:input message="tns:GetSignatureXmlSoapIn" />
  <wsdl:output message="tns:GetSignatureXmlSoapOut" />
</wsdl:operation>
  <wsdl:operation name="RequestClientID">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns an identification node
for this PC.</wsdl:documentation>
  <wsdl:input message="tns:RequestClientIDSoapIn" />
  <wsdl:output message="tns:RequestClientIDSoapOut" />
</wsdl:operation>
  <wsdl:operation name="ServerRequestedWait">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the number of seconds
that the server is requesting the client to wait in order to do load balancing,
etc.</wsdl:documentation>
  <wsdl:input message="tns:ServerRequestedWaitSoapIn" />
  <wsdl:output message="tns:ServerRequestedWaitSoapOut" />
</wsdl:operation>
  <wsdl:operation name="GetDefinitions">
    <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:input message="tns:GetDefinitionsSoapIn" />
  <wsdl:output message="tns:GetDefinitionsSoapOut" />
</wsdl:operation>
  <wsdl:operation name="ReturnResults">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Allows
results to be returned to server from the client</wsdl:documentation>
  <wsdl:input message="tns:ReturnResultsSoapIn" />
  <wsdl:output message="tns:ReturnResultsSoapOut" />
</wsdl:operation>
  <wsdl:operation name="ReturnSystemCharacteristics">
    <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="GetSchema">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Get
the Schemas(XSD) used by the current Schema Version</wsdl:documentation>
        <wsdl:input message="tns:GetSchemaSoapIn" />
        <wsdl:output message="tns:GetSchemaSoapOut" />
    </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: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="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">
    <soap:operation soapAction="http://oval.mitre.org/ServerRequestedWait"
style="document" />
    <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">
    <soap:operation
soapAction="http://oval.mitre.org/ReturnSystemCharacteristics"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetOvalSchemaVersion">
    <soap:operation soapAction="http://oval.mitre.org/GetOvalSchemaVersion"
style="document" />

```

```

    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetSchema">
    <soap:operation soapAction="http://oval.mitre.org/GetSchema"
style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetError">
    <soap:operation soapAction="http://oval.mitre.org/GetError"
style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetSchedule">
    <soap:operation soapAction="http://oval.mitre.org/GetSchedule"
style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="InterpreterServicesSoap12"
type="tns:InterpreterServicesSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="GetCompressionSupported">
    <soap12:operation
soapAction="http://oval.mitre.org/GetCompressionSupported" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RequestSignatureXml">
    <soap12:operation soapAction="http://oval.mitre.org/RequestSignatureXml"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>

```

```

</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:output>
    </wsdl:operation>
    <wsdl:operation name="GetOvalSchemaVersion">
        <soap12:operation
soapAction="http://oval.mitre.org/GetOvalSchemaVersion" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSchema">
        <soap12:operation soapAction="http://oval.mitre.org/GetSchema"
style="document" />
        <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 name="InterpreterServices">
    <wsdl:port name="InterpreterServicesSoap"
binding="tns:InterpreterServicesSoap">
        <soap:address location="http://localhost:1081/Public/IS.asmx" />
    </wsdl:port>
    <wsdl:port name="InterpreterServicesSoap12"
binding="tns:InterpreterServicesSoap12">
        <soap12:address location="http://localhost:1081/Public/IS.asmx" />
    </wsdl:port>
</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.



International | Contact Us | Home

OVAl Service Orientated Architecture

Interpreter Services ▶ CVE Services ▶

Upload History

Computer Name:

Report Uploaded:

Client Time

From: 2006-11-07							To: 2006-12-07						
< November 2006 >							< December 2006 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4	26	27	28	29	30	1	2
5	6	7	8	9	10	11	3	4	5	6	7	8	9
12	13	14	15	16	17	18	10	11	12	13	14	15	16
19	20	21	22	23	24	25	17	18	19	20	21	22	23
26	27	28	29	30	1	2	24	25	26	27	28	29	30
3	4	5	6	7	8	9	31	1	2	3	4	5	6

Click on "[Download](#)" below to download the XML reported

Action	Client Time	Upload At	Schema	Prod Name	Ver	
<input checked="" type="checkbox"/> [Download]	11/14/2006 11:06:25 PM	11/15/2006 9:16:39 PM	5.1	OVAl Definition Interpreter	5.0	44
<input type="checkbox"/> [Download]	11/14/2006 9:07:22 AM	11/16/2006 9:50:21 AM	5.1	OVAl Definition Interpreter	5.1 Build: 14	49
<input type="checkbox"/> [Download]	12/1/2006 1:49:06 PM	12/1/2006 2:20:35 PM	5.1	OVAl Definition Interpreter	5.1 Build: 14	63
<input type="checkbox"/> [Download]	12/1/2006 6:57:23 PM	12/1/2006 6:57:59 PM	5.1	OVAl Definition Interpreter	5.1 Build: 14	68
<input type="checkbox"/> [Download]	12/1/2006 7:02:26 PM	12/1/2006 7:05:01 PM	5.1	OVAl Definition Interpreter	5.1 Build: 14	69
<input checked="" type="checkbox"/> [Download]	12/1/2006 7:05:38 PM	12/1/2006 7:06:13 PM	5.1	OVAl Definition Interpreter	5.1 Build: 14	70
<input checked="" type="checkbox"/> [Download]	12/7/2006 3:35:40 PM	12/7/2006 3:36:20 PM	5.0	sussen	0.33	73
<input checked="" type="checkbox"/> [Download]	12/7/2006 3:36:36 PM	12/7/2006 3:37:00 PM	5.0	sussen	0.33	74

[Download delta report in XML](#)

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>
```

```

    <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 result="error" version="1" reportid="44"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  <definition result="false" version="1" reportid="70"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  <definition version="1" result="false" reportid="73"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  <definition version="1" result="false" reportid="74"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  </element>
  <element id="oval:org.mitre.oval:tst:35">
  <test result="false" version="1" reportid="70"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  <test version="1" result="true" reportid="73"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  </element>
  <element id="oval:org.mitre.oval:tst:57">
  <test result="false" version="1" reportid="70"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  <test version="1" result="true" reportid="73"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  </element>
  <element id="oval:org.mitre.oval:tst:30">
  <test result="true" version="1" reportid="70"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  <test version="1" result="false" reportid="73"
xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" />
  </element>

```


10. Revision History

Version	Date	Author(s)	Description
1.0	2006-10-10	Ken Lassenen	<ul style="list-style-type: none">• Initial Draft for public circulation
1.0.1	2006-11-01	Loren Bandiera Ken Lassenen	<ul style="list-style-type: none">• Reconciliation of PatchLink and MMG Security approaches.• Sussen Implementation added
1.0.2	2006-11-16	Loren Bandiera Ken Lassenen	<ul style="list-style-type: none">• Extended for Schema Version Issues• Optional Calls added
1.0.3	2006-12-06	Ken Lassenen	<ul style="list-style-type: none">• Re-organized and parameter consistency issues corrected.• Added example of application.
1.0.4	2006-12-20	Ken Lassenen	<ul style="list-style-type: none">• Add "RequestSignatureXml" to improve security.

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.