

Oval 5.x Services Orientated Architecture: CVE Services Proposal

[OVAL:SOA:CVES]

Ken Lassenen, Patchlink.com

Intellectual Property Statement

PatchLink 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 February 2007	2
2. Objectives	3
3. Common SOA Features.....	5
3.1 Compression (GetCompressionsSupported)	5
3.2 Security (GetSignatureXml).....	5
3.3 Return Values (GetError).....	6
3.4 Notation.....	6
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. CVE Services.....	10
5.1 Example of Data Sets Returned	11
6. Calls and Examples of XML Returned.....	15
6.1 GetAllData.Xml.....	15
6.2 GetClients.Xml	16
6.3 GetClientsByCVEState.Xml.....	17
6.4 GetCVE.Xml.....	18
6.5 GetCVEClientsByState.Xml.....	19
6.6 GetCVEStateSummary.Xml.....	20
6.7 GetCVEStateTestByClient.Xml.....	21
6.7.1 Example from a proprietary system.....	23
6.8 GetCVESummary.Xml.....	23
7. Comparison Application	25
8. WSDL	27
9. Revision History	38

1.1 RSA Expo OVAL Demonstration February 2007

PatchLink will have the interface described in this document (including any subsequent community evolved modifications) implemented by the demonstration. If you wish to have access to the reference implementation, please email Ken.Lassesen@PatchLink.com

A client for this interface is available that supports comparison between multiple CVE web services. For example:

- Service A: OVAL implementation on a set of computers
- Service B: Proprietary implementation on the set of computers

The results from an automated comparison of evaluation results from each implementation should identify inconsistencies (with the ideal result being better consistency and fewer false positives and false negatives) and allow for improved quality control across the community.

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 is address the CVE Implementation [SOA:CVE], that is the communications between some testing Server that communicates with various clients and a data distribution and collection server that presents summaries of information to management and auditors. Other implementations in this SOA include:

- SOA Intrepreter Service – Communications between the interpreter client (results producer and definitions consumer) and a data distribution and collection server **[OVAL:SOA:IS]**
- 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 author's 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. If the assumption that a single vendor was sufficient is accepted, then for Windows, Microsoft should be sufficient. The majority of vendors in this area reject that

assumptions and with that rejection, the logical consequence of needing multiple players becomes self-evident. The question than arises, should the players play co-operatively or competitively? The author believes that it is in the nation's best interest to be co-operative and thus standardization of services in products should be encouraged.

Ken.Lassesen@patchlink.com

DRAFT ONLY

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 shown 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

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

DRAFT ONLY

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>02Lf2sIDpzpXxib9NhD  
YKGJEZZfmAaTfU5IoN9khlQysfUM2oeRJO/PsM6j6YAe0EuGZ+Hm6L4mDBIt9JOBR3  
PHuvsP9YiECovchb2JOxTmIk9wHCRMyBWM/WiuVFwY0y+f25AiguqADt25C4KQxG  
Xr9paWmIUC454mI8fkDf1U=</Modulus><Exponent>AQAB</Exponent><P>7KiX  
+P/3SAiQI/3M1qUkwZkGwpYzPXM54idyk1vVB0FLvHW5bJaDwAZcn1MKzq9fjg2h  
13eDIaEuLsYJSVbvQ==</P><Q>5KmHcD1nRipmhlkKnVAZL1J+egT9dyf4+6K22y0  
UoYcut+m6IKb/qXD6fIiW6OvHB94VLTUTyiI2My2Zb5BfeQ==</Q><DP>axYSxmp  
Uo7S0cQI4KTxK+ItdAryqdfwXfwFTnRIS+TW0fbyS6BcfJ0B9Q1AtIjaPPkkDa5du/a3  
06n5Ge7B+5Q==</DP><DQ>fGxXG1b+NCrsQNyArxsfqSVdVu4SMd1ZWSuwDIFYc  
ZCKdI7DPRII+cBVPKBiZXmugnRw13mTPCEjlvOQEynfGQ==</DQ><InverseQ>GyS  
QNImDUKAwQP/9YgzapTdOZPHGI3wajDmM0P82K06CiRq8lqugm+S42RhjGxCiGA  
gdGp/9iWdnQHXL+m3MA==</InverseQ><D>jKy6SLzg7YM4Mphz3m0rTQTIE8uw  
t9te/CDCPxhy7tw2Vk5Kp0ZjLIDQGjkUFqBJyF72tpNbwpMC7B1gNRBLqUZ0tOAR9M  
7ZAYYOnTlzv0IIszuE1JBW60NzD5IFUWWutFnYMv5FQ0oZHIFo513ajPLGm+aBXAZ7  
qRVINjUKQ8E=</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>
```

DRAFT ONLY

5. CVE Services

The purposes of CVE Services in an application are two fold:

- Provide a mechanism for comparing results from different systems for purposes of quality control and comparison.
- Provide an interface that higher level management applications can use, reducing development costs and increasing interoperability of systems.

For purposes of client identification, a duality is used:

- @primary_host_name – which is the typical computer name
- <system_info> -- which provides more unique information about the computer

All examples are based on a default **OVAL** implementation.

CVEService

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

- [GetAllData](#)
Returns all of the CVE data
- [GetCVE](#)
Returns the description for the specified CVE.
- [GetCVEClientsByState](#)
Returns the CVE and clients pair for CVEs that have a specific state.
- [GetCVEStateSummary](#)
Returns the list of all CVE's and a count of each state.
- [GetCVEStateTestByClient](#)
Returns all CVE,the state and the test used for one client.
- [GetCVESummary](#)
Returns the list of all CVE's with the number of tests for each CVE.
- [GetClients](#)
Returns the identifiers for all clients that there is CVE information about
- [GetClientsByCVEState](#)
Returns list of clients that has a specific state for a CVE.
- [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 inorder to do load balancing, etc

The methods available can be conceptually described as:

- **GetAllData:** Returns CVE, State, Test, Client Identity, time that the test was performed at. This is effectively a data dump into a flat structure of all of the data available through this interface.
- **GetClients:** Returns a list of computer identities that have at least one CVE evaluated on them and when the last evaluation (of any CVE) was done.
 - *At least one* of the following computer identifications must be given:
 - OVAL's "system_info"
 - Computer name (primary_host_name)
- **GetCVESummary:** Returns a list of all CVE that can be tested for by the application and the number of tests.
 - There may be different tests for different operating systems or versions of operating systems, etc.
- **GetCVEStateSummary:** Returns a list of all CVE that have a status and the number of instances that have this status.
 - An instance means a test on a computer returning this value.
 - A computer may have more than one instance.
- **GetCVEClientsByState:** Returns a list of computer identities where a CVE is in the specified state.
 - *At least one* of the following computer identifications must be given:
 - OVAL's "system_info"
 - Computer name (primary_host_name)
- **GetClientsByCVEState:** Returns a list of computer identities where a specific CVE is a specified state.
- **GetCVEStateTestByClient:** Returns a list of CVE and their state *from each test*.
 - The test_id is the test identifier in the application.
- **GetCVE:** Returns the definition of a CVE

5.1 Example of Data Sets Returned

The following are examples of data that could be returned to clarify the above methods

Table 2 GetCVESummary

cve_name	tests
CVE-2002-0012	3
CVE-2002-0013	2
CVE-2002-0018	2
CVE-2002-0020	1
CVE-2002-0022	1
CVE-2002-0023	4

Table 3 GetCVEStatusSummary

cve_name	state	instances
CVE-2003-0476	not applicable	3
CVE-2003-0476	unknown	1
CVE-2003-0501	not applicable	3
CVE-2003-0501	unknown	1
CVE-2003-0525	false	4

Table 4 GetCVEClientsByState

cve_name	primary_host_name	system_info
----------	-------------------	-------------

CVE-2004-1166	pv-ers-back	<system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5"><os_name>Microsoft Windows Server 2003 Standard Edition Service Pack 1</os_name><os_version>5.2.3790</os_version><architecture>INTEL32</architecture><primary_host_name>pv-ers-back</primary_host_name><interfaces><interface><interface_name>Intel 21140-Based PCI Fast Ethernet Adapter (Generic)</interface_name><ip_address>10.12.20.66</ip_address><mac_address>00-03-FF-76-10-65</mac_address></interface><interface><interface_name>MS TCP Loopback interface</interface_name><ip_address>127.0.0.1</ip_address><mac_address /></interface></interfaces></system_info>
CVE-2005-2266	lassepad	<system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5"><os_name>Microsoft Windows XP</os_name><os_version>5.1.2600</os_version><architecture>unknown</architecture><primary_host_name>lassepad</primary_host_name><interfaces><interface><interface_name>Local Area Connection</interface_name><ip_address>192.168.0.8</ip_address><mac_address>00:E0:B8:89:7D:34</mac_address></interface><interface><interface_name>MS TCP Loopback interface</interface_name><ip_address>127.0.0.1</ip_address><mac_address /></interface><interface><interface_name>Teredo Tunneling Pseudo-Interface</interface_name><ip_address>fe80::ffff:ffff:d%7</ip_address><mac_address>FF:FF:FF:FF:FF:FF</mac_address></interface><interface><interface_name>6to4 Pseudo-Interface</interface_name><ip_address>Unknown</ip_address><mac_address>00:E0:B8:89: mac_address</interface><interface><interface_name>Automatic Tunneling Pseudo-Interface</interface_name><ip_address>fe80::5efe:192.168.0.8%2</ip_address><mac_address>C0:A8:00:08</mac_address></interface><interface><interface_name>Loopback Pseudo-Interface</interface_name><ip_address>::1</ip_address><mac_address /></interface></interfaces></system_info>

Table 5 GetClientByCVEState

primary_host_name	system_info
pv-ers-back	<system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5"><os_name>Microsoft Windows Server 2003 Standard Edition Service Pack 1</os_name><os_version>5.2.3790</os_version><architecture>INTEL32</architecture><primary_host_name>pv-ers-back</primary_host_name><interfaces><interface><interface_name>Intel 21140-Based PCI Fast Ethernet Adapter (Generic)</interface_name><ip_address>10.12.20.66</ip_address><mac_address>00-03-FF-76-10-65</mac_address></interface><interface><interface_name>MS TCP Loopback interface</interface_name><ip_address>127.0.0.1</ip_address><mac_address /></interface></interfaces></system_info>
Lassepad	<system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5"><os_name>Microsoft Windows XP</os_name><os_version>5.1.2600</os_version><architecture>unknown</architecture><primary_host_name>lassepad</primary_host_name><interfaces><interface><interface_name>Local Area Connection</interface_name><ip_address>192.168.0.8</ip_address><mac_address>00:E0:B8:89:7D:34</mac_address></interface><interface><interface_name>MS TCP Loopback interface</interface_name><ip_address>127.0.0.1</ip_address><mac_address /></interface><interface><interface_name>Teredo Tunneling Pseudo-Interface</interface_name><ip_address>fe80::ffff:ffff:d%7</ip_address><mac_address>FF:FF:FF:FF:FF:FF</mac_address></interface><interface><interface_name>6to4 Pseudo-Interface</interface_name><ip_address>Unknown</ip_address><mac_address>00:E0:B8:89: mac_address</interface><interface><interface_name>Automatic Tunneling Pseudo-Interface</interface_name><ip_address>fe80::5efe:192.168.0.8%2</ip_address><mac_address>C0:A8:00:08</mac_address></interface><interface><interface_name>Loopback Pseudo-Interface</interface_name><ip_address>::1</ip_address><mac_address /></interface></interfaces></system_info>

Table 6 GetCVEStateTestByClient

cve_name	state	test_id
CVE-2001-0241	false	oval:org.mitre.oval:def:1068
CVE-2001-0333	false	oval:org.mitre.oval:def:1018
CVE-2001-0333	false	oval:org.mitre.oval:def:1051
CVE-2001-0333	false	oval:org.mitre.oval:def:78

CVE-2001-0333	false	oval:org.mitre.oval:def:37
CVE-2001-0339	false	oval:org.mitre.oval:def:1096

Table 7 GetAllData

Cve_name	state	test_id	primary_host_name	system_info	test_timestamp
CVE-2001-0333	false	oval:org.mitre.oval:def:37	dog		2006-10-29 00:50:45.000
CVE-2001-0333	false	oval:org.mitre.oval:def:78	dog		2006-10-29 00:50:45.000
CVE-2001-0333	false	oval:org.mitre.oval:def:1018	dog		2006-10-29 00:50:45.000
CVE-2001-0333	false	oval:org.mitre.oval:def:1051	dog		2006-10-29 00:50:45.000
CVE-2001-0333	false	oval:org.mitre.oval:def:1018	lassepad		2006-09-27 20:48:30.000
CVE-2001-0333	false	oval:org.mitre.oval:def:1051	lassepad		2006-09-27 20:48:30.000
CVE-2001-0333	false	oval:org.mitre.oval:def:37	lassepad		2006-09-27 20:48:30.000
CVE-2001-0333	false	oval:org.mitre.oval:def:78	lassepad		2006-09-27 20:48:30.000

Table 8 GetClients

primary_host_name	last_report_time	system_info
pv-ers-back	2006-11-18 12:44:54.857	<system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5"><os_name>Microsoft Server 2003 Standard Edition Service Pack 1</os_name><os_version>5.2.3790</os_version><architecture>INTEL32</architecture><primary_host_name>pv-ers-back</primary_host_name><interfaces><interface><interface_name>Intel 21140-Based PCI Fast Ethernet (Generic)</interface_name><ip_address>10.12.20.66</ip_address><mac_address>00-03-FF-76-10-65</mac_address></interface><interface><interface_name>MS TCP Loopback interface</interface_name><ip_address>127.0.0.1</ip_address><mac_address /></interface></interfaces></system_info>
lassepad	2006-10-22 22:54:20.120	<system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5"><os_name>Microsoft XP</os_name><os_version>5.1.2600</os_version><architecture>unknown</architecture><primary_host_name>lassepad</primary_host_name><interfaces><interface><interface_name>Local Area Connection</interface_name><ip_address>192.168.0.8</ip_address><mac_address>00:E0:B8:89:7D:3A</mac_address></interface><interface><interface_name>MS TCP Loopback interface</interface_name><ip_address>127.0.0.1</ip_address><mac_address /></interface><interface><interface_name>Teredo Tunneling Pseudo-Interface</interface_name><ip_address>fe80::ffff:ffff:ffff:7</ip_address><mac_address>FF:FF:FF:FF:FF:FF</mac_address></interface><interface><interface_name>6to4 Pseudo-Interface</interface_name><ip_address>Unknown</ip_address><mac_address>00:E0:B8:89</mac_address></interface><interface_name>Automatic Tunneling Pseudo-Interface</interface_name><ip_address>fe80::5efe:192.168.0.8%2</ip_address><mac_address>C0:A8:00:00</mac_address></interface><interface_name>Loopback Pseudo-Interface</interface_name><ip_address>::1</ip_address><mac_address /></interface></interfaces></system_info>

Table 9 GetCVE

```
<entry type="CVE" name="CVE-2002-0012" seq="2002-0012" published="2002-02-13" modified="2005-10-20" severity="High"
  <desc>
    <descript source="cve">Vulnerabilities in a large number of SNMP implementations allow remote attackers to ca
  by the PROTOS c06-SNMPv1 test suite. NOTE: It is highly likely that this candidate will be SPLIT into multiple c
  be updated when more accurate information is available.</descript>
  </desc>
  <loss_types>
    <avail />
    <sec_prot admin="1" />
  </loss_types>
  <vuln_types>
    <exception />
  </vuln_types>
  <range>
    <remote />
  </range>
  <refs>
    <ref source="ISS X-Force" url="http://www.iss.net/security_center/alerts/advisel10.php" adv="1">ISS:20020212
    <ref source="CERT" url="http://www.cert.org/advisories/CA-2002-03.html" adv="1">CERT:CA-2002-03</ref>
    <ref source="CERT" url="http://www.kb.cert.org/vuls/id/107186" adv="1">CERT-VN:VU#107186</ref>
    <ref source="MISC" url="http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/index.html">http://www
    <ref source="REDHAT" url="http://www.redhat.com/support/errata/RHSA-2001-163.html" adv="1">RHSA-2001:163</ref>
    <ref source="SGI" url="ftp://patches.sgi.com/support/free/security/advisories/20020201-01-A" adv="1" patch="1"
    <ref source="MS" url="http://www.microsoft.com/technet/security/bulletin/MS02-006.asp" adv="1" patch="1">MS02
    <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval144.html" sig="1">OVAL144</ref>
    <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval161.html" sig="1">OVAL161</ref>
    <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval298.html" sig="1">OVAL298</ref>
    <ref source="HP" url="http://www.securityfocus.com/advisories/4211" adv="1" patch="1">HPSBMP0206-015</ref>
    <ref source="BIID" url="http://www.securityfocus.com/bid/5043" patch="1">5043</ref>
    <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval1048.html" sig="1">OVAL1048</ref>
    <ref source="OVAL" url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:144" sig="1">
    <ref source="OVAL" url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:161" sig="1">
    <ref source="OVAL" url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:298" sig="1">
    <ref source="OVAL" url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:1048" sig="1">
  </refs>
  <vuln_soft>
    <prod name="SNMP" vendor="SNMP">
      <vers num="" />
    </prod>
  </vuln_soft>
</entry>
```

DRAFT

6. Calls and Examples of XML Returned

All of the XML implements the following pattern:

- The method name is the root element name.
- The time (utc) of generation is recorded in the root element (@utctime)
- Any filtering parameters is echoed in the root element (or the first child if system_info)
- The system/software producing the information is recorded in the root element (@source)

6.1 GetAllData.Xml

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

DRAFT ONLY

```

<?xml version="1.0" standalone="yes"?>
<GetAllData source="OVAL Add-In" utctime="2006-12-01T19:06:51">
  <entry cve_name="CVE-1999-0278" state="not applicable"
test_id="oval:org.mitre.oval:def:913" primary_host_name="dog"
test_timestamp="2006-12-01T09:44:58">
    <system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">
      <os_name>Microsoft Windows Server 2003 Professional Service Pack
1</os_name>
      <os_version>5.2.3790</os_version>
      <architecture>INTEL32</architecture>
      <primary_host_name>dog</primary_host_name>
      <interfaces>
        <interface>
          <interface_name>NVIDIA nForce Networking Controller - Packet
Scheduler Miniport</interface_name>
          <ip_address>192.168.0.251</ip_address>
          <mac_address>00-17-31-36-0F-C8</mac_address>
        </interface>
        <interface>
          <interface_name>MS TCP Loopback interface</interface_name>
          <ip_address>127.0.0.1</ip_address>
          <mac_address />
        </interface>
      </interfaces>
    </system_info>
  </entry>
  <entry cve_name="CVE-1999-0562" state="not applicable"
test_id="oval:org.mitre.oval:def:1023" primary_host_name="dog"
test_timestamp="2006-12-01T09:44:58">
    <system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">
      <os_name>Microsoft Windows Server 2003 Professional Service Pack
1</os_name>
      <os_version>5.2.3790</os_version>
      <architecture>INTEL32</architecture>
      <primary_host_name>dog</primary_host_name>
      <interfaces>
        <interface>
          <interface_name>NVIDIA nForce Networking Controller - Packet
Scheduler Miniport</interface_name>
          <ip_address>192.168.0.251</ip_address>
          <mac_address>00-17-31-36-0F-C8</mac_address>
        </interface>
        <interface>
          <interface_name>MS TCP Loopback interface</interface_name>
          <ip_address>127.0.0.1</ip_address>
          <mac_address />
        </interface>
      </interfaces>
    </system_info>
  </entry>
</GetAllData>

```

6.2 GetClients.Xml

```

public int GetClients (
    string compressionType,

```

```
out byte [ ] data )
```

```
<?xml version="1.0" standalone="yes"?>
<GetClients source="OVAL Add-In" utctime="2006-11-29T18:25:45">
  <entry primary_host_name="lassepad" last_report_time="2006-11-18T12:44:54">
    <system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">
      . . .
    </system_info>
  </entry>
  <entry primary_host_name="dog" last_report_time="2006-11-25T22:33:54">
    <system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">
      . . .
    </system_info>
  </entry>
  <entry primary_host_name="Lasseesen" last_report_time="2006-10-22T22:54:20">
    . . .
  </entry>
</GetClients>
```

6.3 GetClientsByCVEState.Xml

```
public int GetClientsByCVEState(
    string compressionType,
    string cve_name,
    string state,
    out byte[] data)
```

```
<?xml version="1.0" standalone="yes"?>
<GetClientsByCVEState source="OVAL Add-In" cve_name="CVE-2002-0012"
state="false" utctime="2006-11-29T18:25:46">
  <entry primary_host_name="dog">
    <system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">
      <os_name>Microsoft Windows Server 2003 Professional Service Pack
1</os_name>
      <os_version>5.2.3790</os_version>
      <architecture>INTEL32</architecture>
      <primary_host_name>dog</primary_host_name>
      <interfaces>
        <interface>
          <interface_name>NVIDIA nForce Networking Controller - Packet
Scheduler Miniport</interface_name>
          <ip_address>192.168.0.251</ip_address>
          <mac_address>00-17-31-36-0F-C8</mac_address>
        </interface>
        <interface>
          <interface_name>MS TCP Loopback interface</interface_name>
          <ip_address>127.0.0.1</ip_address>
          <mac_address />
        </interface>
      </interfaces>
    </system_info>
  </entry>
  <entry primary_host_name="dog">
    . . .
  </entry>
</GetClientsByCVEState>
```

6.4 GetCVE.Xml

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

DRAFT ONLY

```

<?xml version="1.0" standalone="yes"?>
<GetCVE source="OVAL Add-In" utctime="2006-11-29T18:25:46" cve_name="CVE-2002-0012">
  <entry type="CVE" name="CVE-2002-0012" seq="2002-0012" published="2002-02-13" modified="2005-
10-20" severity="High" CVSS_score="10.0" CVSS_vector="(AV:R/AC:L/Au:NR/C:C/I:C/A:C/B:N)
Approximated">
    <desc>
      <descript source="cve">Vulnerabilities in a large number of SNMP implementations allow
remote attackers to cause a denial of service or gain privileges via SNMPv1 trap handling, as
demonstrated by the PROTOS c06-SNMPv1 test suite. NOTE: It is highly likely that this candidate
will be SPLIT into multiple candidates, one or more for each vendor. This and other SNMP-related
candidates will be updated when more accurate information is available.</descript>
    </desc>
    <loss_types>
      <avail />
      <sec_prot admin="1" />
    </loss_types>
    <vuln_types>
      <exception />
    </vuln_types>
    <range>
      <remote />
    </range>
    <refs>
      <ref source="ISS X-Force" url="http://www.iss.net/security_center/alerts/advisel10.php"
adv="1">ISS:20020212 PROTOS Remote SNMP Attack Tool</ref>
      <ref source="CERT" url="http://www.cert.org/advisories/CA-2002-03.html" adv="1">CERT:CA-
2002-03</ref>
      <ref source="CERT" url="http://www.kb.cert.org/vuls/id/107186" adv="1">CERT-
VN:VU#107186</ref>
      <ref source="MISC"
url="http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/index.html">http://www.ee.oul
u.fi/research/ouspg/protos/testing/c06/snmpv1/index.html</ref>
      <ref source="REDHAT" url="http://www.redhat.com/support/errata/RHSA-2001-163.html"
adv="1">RHSA-2001:163</ref>
      <ref source="SGI" url="ftp://patches.sgi.com/support/free/security/advisories/20020201-01-
A" adv="1" patch="1">20020201-01-A</ref>
      <ref source="MS" url="http://www.microsoft.com/technet/security/bulletin/MS02-006.asp"
adv="1" patch="1">MS02-006</ref>
      <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval144.html"
sig="1">OVAL144</ref>
      <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval161.html"
sig="1">OVAL161</ref>
      <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval298.html"
sig="1">OVAL298</ref>
      <ref source="HP" url="http://www.securityfocus.com/advisories/4211" adv="1"
patch="1">HPSBMP0206-015</ref>
      <ref source="BID" url="http://www.securityfocus.com/bid/5043" patch="1">5043</ref>
      <ref source="OVAL" url="http://oval.mitre.org/oval/definitions/data/oval1048.html"
sig="1">OVAL1048</ref>
      <ref source="OVAL"
url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:144"
sig="1">oval:org.mitre.oval:def:144</ref>
      <ref source="OVAL"
url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:161"
sig="1">oval:org.mitre.oval:def:161</ref>
      <ref source="OVAL"
url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:298"
sig="1">oval:org.mitre.oval:def:298</ref>
      <ref source="OVAL"
url="http://oval.mitre.org/repository/data/getDef?id=oval:org.mitre.oval:def:1048"
sig="1">oval:org.mitre.oval:def:1048</ref>
    </refs>
    <vuln_soft>
      <prod name="SNMP" vendor="SNMP">
        <vers num="" />
      </prod>
    </vuln_soft>
  </entry>
</GetCVE>

```

6.5 GetCVEClientsByState.Xml

```
public int GetCVEClientsByState(
```

```
string compressionType,  
string state,  
out byte[] data)
```

```
<?xml version="1.0" standalone="yes"?>  
<GetCVEClientsByState source="OVAL Add-In" utctime="2006-11-29T18:25:46"  
state="true">  
  <entry cve_name="CVE-2004-1166" primary_host_name="pv-ers-back">  
    <system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-  
characteristics-5">  
      <os_name>Microsoft Windows Server 2003 Standard Edition Service Pack  
1</os_name>  
      <os_version>5.2.3790</os_version>  
      <architecture>INTEL32</architecture>  
      <primary_host_name>pv-ers-back</primary_host_name>  
      <interfaces>  
        <interface>  
          <interface_name>Intel 21140-Based PCI Fast Ethernet Adapter  
(Generic)</interface_name>  
          <ip_address>10.12.20.66</ip_address>  
          <mac_address>00-03-FF-76-10-65</mac_address>  
        </interface>  
        <interface>  
          <interface_name>MS TCP Loopback interface</interface_name>  
          <ip_address>127.0.0.1</ip_address>  
          <mac_address />  
        </interface>  
      </interfaces>  
    </system_info>  
  </entry>  
  <entry cve_name="CVE-2005-2266" primary_host_name="lassepad">  
    <system_info xmlns="http://oval.mitre.org/XMLSchema/oval-system-  
characteristics-5">  
      .  
      .  
      .  
    </system_info>  
  </entry>  
</GetCVEClientsByState>
```

6.6 GetCVEStateSummary.Xml

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

```

<?xml version="1.0" standalone="yes"?>
<GetCVEStateSummary source="OVAL Add-In" utctime="2006-11-29T18:25:46">
  <entry cve_name="CVE-1999-0278" state="false" instances="4" />
  <entry cve_name="CVE-1999-0562" state="false" instances="4" />
  <entry cve_name="CVE-1999-0621" state="false" instances="4" />
  <entry cve_name="CVE-1999-0689" state="not applicable" instances="3" />
  <entry cve_name="CVE-1999-0689" state="unknown" instances="1" />
  <entry cve_name="CVE-1999-0691" state="not applicable" instances="3" />
  <entry cve_name="CVE-1999-0691" state="unknown" instances="1" />
  <entry cve_name="CVE-1999-0693" state="not applicable" instances="3" />
  <entry cve_name="CVE-1999-0693" state="unknown" instances="1" />
  <entry cve_name="CVE-1999-0736" state="false" instances="4" />
  <entry cve_name="CVE-1999-0815" state="false" instances="4" />
  <entry cve_name="CVE-1999-0874" state="false" instances="4" />
  <entry cve_name="CVE-2000-0377" state="false" instances="4" />
  <entry cve_name="CVE-2000-0778" state="false" instances="4" />
  <entry cve_name="CVE-2000-0884" state="false" instances="4" />
  <entry cve_name="CVE-2000-0886" state="false" instances="4" />
  <entry cve_name="CVE-2000-0979" state="false" instances="4" />
  <entry cve_name="CVE-2000-1079" state="false" instances="4" />
  <entry cve_name="CVE-2000-1081" state="false" instances="4" />
  <entry cve_name="CVE-2000-1134" state="not applicable" instances="3" />
  <entry cve_name="CVE-2000-1134" state="unknown" instances="1" />
  <entry cve_name="CVE-2001-0002" state="false" instances="4" />
  <entry cve_name="CVE-2001-0045" state="false" instances="4" />
  <entry cve_name="CVE-2001-0046" state="false" instances="4" />
  <entry cve_name="CVE-2001-0047" state="false" instances="4" />
  <entry cve_name="CVE-2001-0151" state="false" instances="4" />
  <entry cve_name="CVE-2001-0154" state="false" instances="4" />
</GetCVEStateSummary>

```

6.7 GetCVEStateTestByClient.Xml

```

public int GetCVEStateTestByClient(
    string compressionType,
    string primary_host_name,
    byte[] system_info,
    out byte[] data)

```

```

<?xml version="1.0" standalone="yes"?>
<GetCVEStateTestByClient source="OVAL Add-In" utctime="2006-11-29T20:39:41"
primary_host_name="dog">
  <os_name xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5">Microsoft Windows Server 2003 Professional Service Pack 1</os_name>
  <os_version xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5">5.2.3790</os_version>
  <architecture xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5">INTEL32</architecture>
  <primary_host_name xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5">dog</primary_host_name>
  <interfaces xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5">
    <interface>
      <interface_name>NVIDIA nForce Networking Controller - Packet Scheduler Miniport</interface_name>
      <ip_address>192.168.0.251</ip_address>
      <mac_address>00-17-31-36-0F-C8</mac_address>
    </interface>
    <interface>
      <interface_name>MS TCP Loopback interface</interface_name>
      <ip_address>127.0.0.1</ip_address>
      <mac_address />
    </interface>
  </interfaces>
  <host_identification url="localhost">493fd249-c09d-4dbb-aeda-3ab3b8685753</host_identification>
  <entry cve_name="CVE-1999-0278" state="false"
test_id="oval:org.mitre.oval:def:913" />
  <entry cve_name="CVE-1999-0562" state="false"
test_id="oval:org.mitre.oval:def:1023" />
  <entry cve_name="CVE-1999-0621" state="false"
test_id="oval:org.mitre.oval:def:1024" />
  <entry cve_name="CVE-1999-0689" state="not applicable"
test_id="oval:org.mitre.oval:def:1880" />
  <entry cve_name="CVE-1999-0691" state="not applicable"
test_id="oval:org.mitre.oval:def:3078" />
  <entry cve_name="CVE-1999-0693" state="not applicable"
test_id="oval:org.mitre.oval:def:4374" />
  <entry cve_name="CVE-1999-0736" state="false"
test_id="oval:org.mitre.oval:def:932" />
  .
  .
  .
</GetCVEStateTestByClient>

```

6.7.1 Example from a proprietary system

```
<?xml version="1.0" standalone="yes"?>
<GetCVEStateTestByClient source="PatchLink Plus" utctime="2006-12-01T20:02:59"
primary_host_name="dog.corporate.patchlink">
  <system_info>
    <os_name xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">Microsoft Windows Server 2003 Professional Service Pack
1</os_name>
    <os_version xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">5.2.3790</os_version>
    <architecture xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">INTEL32</architecture>
    <primary_host_name xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">dog.corporate.patchlink</primary_host_name>
    <interfaces xmlns="http://oval.mitre.org/XMLSchema/oval-system-
characteristics-5">
      <interface>
        <interface_name>NVIDIA nForce Networking Controller - Packet Scheduler
Miniport</interface_name>
        <ip_address>192.168.0.251</ip_address>
        <mac_address>00-17-31-36-0F-C8</mac_address>
      </interface>
      <interface>
        <interface_name>MS TCP Loopback interface</interface_name>
        <ip_address>127.0.0.1</ip_address>
        <mac_address />
      </interface>
    </interfaces>
    <host_identification url="localhost">493fd249-c09d-4dbb-aeda-
3ab3b8685753</host_identification>
  </system_info>
  <entry cve_name="CVE-1999-0008" state="false" test_id="2F074649-EC41-4E6B-
864D-14376F05A58A" />
  <entry cve_name="CVE-1999-0008" state="false" test_id="22CCCA98-FB5C-4F03-
AFD8-375248B5AB28" />
  <entry cve_name="CVE-1999-0009" state="false" test_id="688EC8BD-897B-45C2-
863E-ECE089AE9F1E" />
  <entry cve_name="CVE-1999-0011" state="false" test_id="688EC8BD-897B-45C2-
863E-ECE089AE9F1E" />
  <entry cve_name="CVE-1999-0054" state="false" test_id="36BDF1BB-9A16-4512-
B6A0-B14DDC0CCDC7" />
  <entry cve_name="CVE-1999-0693" state="false" test_id="461CDE48-2021-4CA2-
B388-CBDE692E3C6B" />
  <entry cve_name="CVE-1999-0693" state="false" test_id="2C5CD6DA-31B1-40FC-
ABD2-D262F284327A" />
  <entry cve_name="CVE-1999-0696" state="false" test_id="6AA3ADE4-D7BF-4103-
8323-50B57DB4DBD5" />
  .
  .
  .
</GetCVEStateTestByClient>
```

6.8 GetCVESummary.Xml

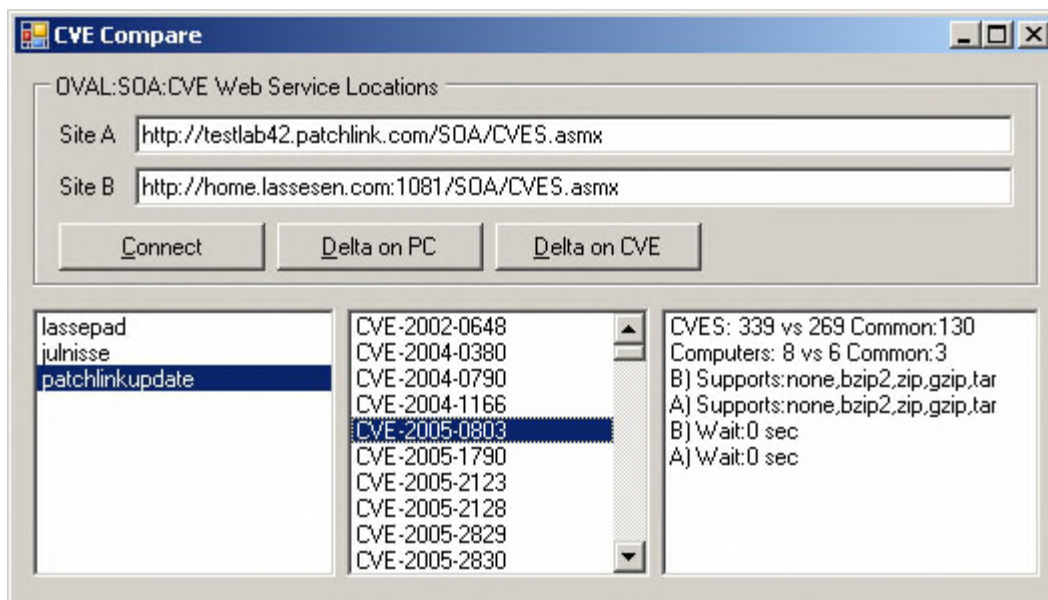
```
public int GetCVESummary(
    string compressionType,
    out byte[] data)
```

```
<?xml version="1.0" standalone="yes"?>
<GetCVESummary source="OVAL Add-In" utctime="2006-11-29T18:25:46">
  <entry cve_name="CVE-1999-0278" tests="1" />
  <entry cve_name="CVE-1999-0562" tests="1" />
  <entry cve_name="CVE-1999-0621" tests="1" />
  <entry cve_name="CVE-1999-0689" tests="1" />
  <entry cve_name="CVE-1999-0691" tests="1" />
  <entry cve_name="CVE-1999-0693" tests="1" />
  <entry cve_name="CVE-1999-0736" tests="1" />
  <entry cve_name="CVE-1999-0815" tests="1" />
  <entry cve_name="CVE-1999-0874" tests="1" />
  <entry cve_name="CVE-2000-0377" tests="1" />
  <entry cve_name="CVE-2000-0778" tests="1" />
  <entry cve_name="CVE-2000-0884" tests="1" />
  <entry cve_name="CVE-2000-0886" tests="1" />
  <entry cve_name="CVE-2000-0979" tests="1" />
  <entry cve_name="CVE-2000-1079" tests="1" />
  <entry cve_name="CVE-2000-1081" tests="1" />
  <entry cve_name="CVE-2000-1134" tests="1" />
  <entry cve_name="CVE-2001-0002" tests="1" />
  <entry cve_name="CVE-2001-0045" tests="1" />
  <entry cve_name="CVE-2001-0046" tests="1" />
  <entry cve_name="CVE-2001-0047" tests="1" />
  <entry cve_name="CVE-2001-0151" tests="1" />
  <entry cve_name="CVE-2001-0154" tests="1" />
  <entry cve_name="CVE-2001-0241" tests="1" />
  <entry cve_name="CVE-2001-0333" tests="4" />
  <entry cve_name="CVE-2001-0339" tests="1" />
  <entry cve_name="CVE-2001-0344" tests="1" />
  <entry cve_name="CVE-2001-0414" tests="1" />
  .
  .
  .
</GetCVESummary>
```

DRAFT

7. Comparison Application

A simple application has been created that allows two instances of this interface to be compared.



After the initial connection, the computers that exist on both systems are listed on the left and the CVE's that are in common is listed in the middle.

In many cases, it is expected that there will not be an exact match. As can be seen above, one system has 209 CVE's (339-120) that are not available on the other system, and the other system has 139 CVE's missing from the other. There are 130 matches.

The above application produces XML files such as the one shown below.

```
<cve_delta pc="patchlinkupdate">
  <entry cve_name="CVE-2002-0648">
    <entry state="false" test_id="F7DD139B-7515-4E5B-9D30-DA1048562123" />
    <entry state="false" test_id="oval:org.mitre.oval:def:608" />
  </entry>
  <entry cve_name="CVE-2004-0380">
    <entry state="false" test_id="0E65238A-8962-4BAA-ADF1-0DBF1D5DB766" />
    <entry state="false" test_id="oval:org.mitre.oval:def:1028" />
  </entry>
  <entry cve_name="CVE-2004-0790">
    <entry state="false" test_id="8FCE3929-E649-4F81-B9EC-AA08E301621A" />
    <entry state="false" test_id="oval:org.mitre.oval:def:4804" />
  </entry>
  . . .
</cve_delta>
```

In some cases, one system may test for a CVE with multiple tests. The first example shows duplicate for a proprietary system.

```
<entry cve_name="CVE-2004-1166">
  <entry state="false" test_id="97E8EAEB-D373-463D-BE47-6B2ABB9730D1" />
  <entry state="false" test_id="6A6AB610-A9E2-47C0-97B8-B42DC3ACC1DA" />
  <entry state="false" test_id="oval:org.mitre.oval:def:462" />
</entry>
```

The second example shows duplicate for OVAL

```
<entry cve_name="CVE-2005-0803">
  <entry state="false" test_id="CC9A3781-8F58-475A-876B-85E4A902AC4D" />
  <entry state="false" test_id="oval:org.mitre.oval:def:1215" />
  <entry state="false" test_id="oval:org.mitre.oval:def:671" />
</entry>
```

There can be cases where there is disagreement between tests. Whether this is “wrong” or not, depends on the nature of the tests. For example, the proprietary system below may require all tests to be “true” for the result to be “true”.

```
<entry cve_name="CVE-2005-4560">
  <entry state="true" test_id="BE31F161-36A6-4F7A-BE1B-22C0D327088C" />
  <entry state="false" test_id="A9A46C35-A867-4CE7-8843-F3370C7FF959" />
  <entry state="true" test_id="C6265CB3-DA88-4E90-BC70-F4EA4A241D58" />
  <entry state="false" test_id="oval:org.mitre.oval:def:1612" />
</entry>
```

For OVAL tests, it also can be used to determine consistency when there are multiple tests for the one CVE (remember that test_id here is a reference to a definition and not a OVAL_test).

```
<entry cve_name="CVE-2006-0002">
  <entry state="false" test_id="oval:org.mitre.oval:def:624" />
  <entry state="false" test_id="oval:org.mitre.oval:def:1082" />
  <entry state="false" test_id="oval:org.mitre.oval:def:1456" />
  <entry state="false" test_id="oval:org.mitre.oval:def:1485" />
  <entry state="false" test_id="oval:org.mitre.oval:def:1316" />
  <entry state="false" test_id="oval:org.mitre.oval:def:1165" />
</entry>
```

Most firms issue contents on Patch Tuesday and OVAL definitions are also issue at the same time. Using this tool allows reconciliation to be done quickly and improve quality control for both OVAL and for proprietary systems.

8. WSDL

```
<?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://cve.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://cve.mitre.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://cve.mitre.org/">
      <s:element name="GetCVESummary">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetCVESummaryResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetCVESummaryResult"
type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetCVEStateSummary">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetCVEStateSummaryResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1"
name="GetCVEStateSummaryResult" type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetCVEClientsByState">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="state"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>

```

```

    </s:complexType>
  </s:element>
  <s:element name="GetCVEClientsByStateResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1"
name="GetCVEClientsByStateResult" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetClientsByCVEState">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="cve_name"
type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="state"
type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetClientsByCVEStateResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1"
name="GetClientsByCVEStateResult" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetCVEStateTestByClient">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="primary_host_name"
type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="system_info"
type="s:base64Binary" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetCVEStateTestByClientResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1"
name="GetCVEStateTestByClientResult" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetAllData">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />

```

```

        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetAllDataResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetAllDataResult"
type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetClients">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetClientsResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetClientsResult"
type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetCVE">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="compressionType"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="cve_name"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetCVEResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetCVEResult"
type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="data"
type="s:base64Binary" />
        </s:sequence>
    </s:complexType>
</s:element>
<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="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="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="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:schema>
</wsdl:types>
<wsdl:message name="GetCVESummarySoapIn">
  <wsdl:part name="parameters" element="tns:GetCVESummary" />
</wsdl:message>
<wsdl:message name="GetCVESummarySoapOut">
  <wsdl:part name="parameters" element="tns:GetCVESummaryResponse" />
</wsdl:message>
<wsdl:message name="GetCVEStateSummarySoapIn">
  <wsdl:part name="parameters" element="tns:GetCVEStateSummary" />
</wsdl:message>
<wsdl:message name="GetCVEStateSummarySoapOut">
  <wsdl:part name="parameters" element="tns:GetCVEStateSummaryResponse" />
</wsdl:message>
<wsdl:message name="GetCVEClientsByStateSoapIn">

```

```

    <wsdl:part name="parameters" element="tns:GetCVEClientsByState" />
</wsdl:message>
<wsdl:message name="GetCVEClientsByStateSoapOut">
    <wsdl:part name="parameters" element="tns:GetCVEClientsByStateResponse" />
</wsdl:message>
<wsdl:message name="GetClientsByCVEStateSoapIn">
    <wsdl:part name="parameters" element="tns:GetClientsByCVEState" />
</wsdl:message>
<wsdl:message name="GetClientsByCVEStateSoapOut">
    <wsdl:part name="parameters" element="tns:GetClientsByCVEStateResponse" />
</wsdl:message>
<wsdl:message name="GetCVEStateTestByClientSoapIn">
    <wsdl:part name="parameters" element="tns:GetCVEStateTestByClient" />
</wsdl:message>
<wsdl:message name="GetCVEStateTestByClientSoapOut">
    <wsdl:part name="parameters" element="tns:GetCVEStateTestByClientResponse"
/>
</wsdl:message>
<wsdl:message name="GetAllDataSoapIn">
    <wsdl:part name="parameters" element="tns:GetAllData" />
</wsdl:message>
<wsdl:message name="GetAllDataSoapOut">
    <wsdl:part name="parameters" element="tns:GetAllDataResponse" />
</wsdl:message>
<wsdl:message name="GetClientsSoapIn">
    <wsdl:part name="parameters" element="tns:GetClients" />
</wsdl:message>
<wsdl:message name="GetClientsSoapOut">
    <wsdl:part name="parameters" element="tns:GetClientsResponse" />
</wsdl:message>
<wsdl:message name="GetCVESoapIn">
    <wsdl:part name="parameters" element="tns:GetCVE" />
</wsdl:message>
<wsdl:message name="GetCVESoapOut">
    <wsdl:part name="parameters" element="tns:GetCVEResponse" />
</wsdl:message>
<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="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="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="GetErrorSoapIn">
    <wsdl:part name="parameters" element="tns:GetError" />
</wsdl:message>
<wsdl:message name="GetErrorSoapOut">
    <wsdl:part name="parameters" element="tns:GetErrorResponse" />
</wsdl:message>

```

```

<wsdl:portType name="OVAL_CVEServiceSoap">
  <wsdl:operation name="GetCVESummary">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the list of all CVE's
with the number of tests for each CVE.</wsdl:documentation>
    <wsdl:input message="tns:GetCVESummarySoapIn" />
    <wsdl:output message="tns:GetCVESummarySoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetCVEStateSummary">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the list of all CVE's
and a count of each state.</wsdl:documentation>
    <wsdl:input message="tns:GetCVEStateSummarySoapIn" />
    <wsdl:output message="tns:GetCVEStateSummarySoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetCVEClientsByState">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the CVE and clients pair
for CVEs that have a specific state.</wsdl:documentation>
    <wsdl:input message="tns:GetCVEClientsByStateSoapIn" />
    <wsdl:output message="tns:GetCVEClientsByStateSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetClientsByCVEState">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns list of clients that has
a specific state for a CVE.</wsdl:documentation>
    <wsdl:input message="tns:GetClientsByCVEStateSoapIn" />
    <wsdl:output message="tns:GetClientsByCVEStateSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetCVEStateTestByClient">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns all CVE,the state and
the test used for one client.</wsdl:documentation>
    <wsdl:input message="tns:GetCVEStateTestByClientSoapIn" />
    <wsdl:output message="tns:GetCVEStateTestByClientSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetAllData">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns all of the CVE
data</wsdl:documentation>
    <wsdl:input message="tns:GetAllDataSoapIn" />
    <wsdl:output message="tns:GetAllDataSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetClients">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the identifiers for all
clients that there is CVE information about </wsdl:documentation>
    <wsdl:input message="tns:GetClientsSoapIn" />
    <wsdl:output message="tns:GetClientsSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetCVE">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the description for the
specified CVE.</wsdl:documentation>
    <wsdl:input message="tns:GetCVESoapIn" />
    <wsdl:output message="tns:GetCVESoapOut" />
  </wsdl:operation>
  <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="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="ServerRequestedWait">
        <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:input message="tns:ServerRequestedWaitSoapIn" />
        <wsdl:output message="tns:ServerRequestedWaitSoapOut" />
    </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:portType>
<wsdl:binding name="OVAL_CVEServiceSoap" type="tns:OVAL_CVEServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetCVESummary">
        <soap:operation soapAction="http://cve.mitre.org/GetCVESummary"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCVEStateSummary">
        <soap:operation soapAction="http://cve.mitre.org/GetCVEStateSummary"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCVEClientsByState">
        <soap:operation soapAction="http://cve.mitre.org/GetCVEClientsByState"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetClientsByCVEState">

```

```

        <soap:operation soapAction="http://cve.mitre.org/GetClientsByCVEState"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCVEStateTestByClient">
        <soap:operation
soapAction="http://cve.mitre.org/GetCVEStateTestByClient" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAllData">
        <soap:operation soapAction="http://cve.mitre.org/GetAllData"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetClients">
        <soap:operation soapAction="http://cve.mitre.org/GetClients"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCVE">
        <soap:operation soapAction="http://cve.mitre.org/GetCVE"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCompressionSupported">
        <soap:operation
soapAction="http://cve.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="GetSignatureXml">

```

```

        <soap:operation soapAction="http://cve.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="ServerRequestedWait">
        <soap:operation soapAction="http://cve.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="GetError">
        <soap:operation soapAction="http://cve.mitre.org/GetError"
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="OVAL_CVEServiceSoap12" type="tns:OVAL_CVEServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetCVESummary">
        <soap12:operation soapAction="http://cve.mitre.org/GetCVESummary"
style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCVEStateSummary">
        <soap12:operation soapAction="http://cve.mitre.org/GetCVEStateSummary"
style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCVEClientsByState">
        <soap12:operation soapAction="http://cve.mitre.org/GetCVEClientsByState"
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="GetClientsByCVEState">
  <soap12:operation soapAction="http://cve.mitre.org/GetClientsByCVEState"
style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCVEStateTestByClient">
  <soap12:operation
soapAction="http://cve.mitre.org/GetCVEStateTestByClient" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAllData">
  <soap12:operation soapAction="http://cve.mitre.org/GetAllData"
style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetClients">
  <soap12:operation soapAction="http://cve.mitre.org/GetClients"
style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCVE">
  <soap12:operation soapAction="http://cve.mitre.org/GetCVE"
style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCompressionSupported">
  <soap12:operation
soapAction="http://cve.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="GetSignatureXml">
      <soap12:operation soapAction="http://cve.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="ServerRequestedWait">
      <soap12:operation soapAction="http://cve.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="GetError">
      <soap12:operation soapAction="http://cve.mitre.org/GetError"
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="OVAL_CVEService">
    <wsdl:port name="OVAL_CVEServiceSoap" binding="tns:OVAL_CVEServiceSoap">
      <soap:address location="http://localhost:1081/Testing/CVES.asmx" />
    </wsdl:port>
    <wsdl:port name="OVAL_CVEServiceSoap12"
binding="tns:OVAL_CVEServiceSoap12">
      <soap12:address location="http://localhost:1081/Testing/CVES.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

DRAFT

9. Revision History

Version	Date	Author(s)	Description
1.0	2006-12-08	Ken Lassenen	<ul style="list-style-type: none">• Initial Draft for public circulation
			<ul style="list-style-type: none">•

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.

DRAFT ONLY