OVAL 5 Tutorial

by Matt Payne & Jason Smith

Slides online at:
OVALtools.org

Your Key to Security

About

- This talk
  - Draws from
    - NIST talks
    - OVAL XML Schema's internal documentation
    - OVAL MITRE.org webpages
  - Prepared by Jason Smith & Matt Payne
- This speaker
  - Matt Payne, CISSP & NUCIA.UNOmaha.edu Research Fellow… .OVALtools.org & MattPayne.org

Agenda

- Conceptual overview from Stephen Quinn & Peter Mell's NIST briefing:
  - Automated Security Measurement/ FISMA Technical Control Automation
- **Components of OVAL:**
  - **Language, Repositories, MITRE's Interpreter**
- **Getting started with MITRE's Interpeter**
- **Overview of OVAL tests**
- **Advertisement for OVALtools.org's OVALdsl**
- **Advertisement for what you can do!**
- **What's next?**

Gifts to Community

COTS Tool Vendors –

- Provision of an enhanced IT security data repository
  - No cost and license free
  - CVE/OVAL/XCCDF/CVSS
  - Cover both patches and configuration issues
- Elimination of duplication of effort
- Cost reduction through standardization

Federal Agencies

- Automation of technical control compliance (FISMA)

Current Problems

Conceptual Analogy

Current Problems

Conceptual Analogy Continued (2)

Outsource

Ford

PORSCHE

PORSCHE

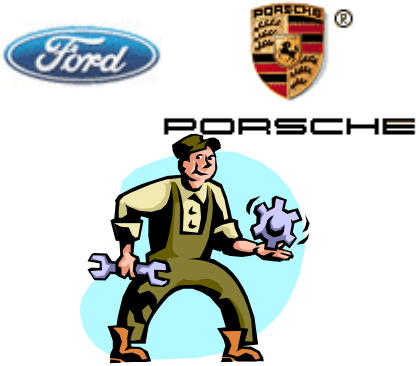
ASE National Institute for AUTOMOTIVE SERVICE EXCELLENCE

In-House

Current Problems

Conceptual Analogy Continued (3)

Outsource



In-House

a.) Troubleshoot/Analyze

- Conduct Testing
- Is there a problem?
- Cause of error condition?
- Is this check reporting correctly?

b.) Document/Report Findings

c.) Recommendations

d.) Remediate

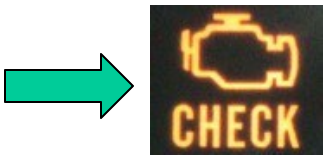
Current Problems

Conceptual Analogy Continued (5)

Standardize & Automate

a.) Troubleshoot/Analyze

- Is there a problem?
- Cause of error condition?
- Is this check reporting correctly?



More DATA

Outsource

In-House

a.) Troubleshoot/Analyze

- Conduct Testing
- Is there a problem?
- Cause of error condition?
- Is this check reporting correctly?

b.) Document/Report Findings

c.) Recommendations

d.) Remediate

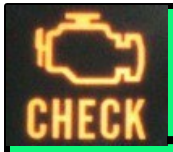
Current Problems

Conceptual Analogy Continued (6)

Your Key to Security

Before

After



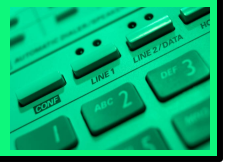
Error Report

Problem:
Air Pressure Loss

Diagnosis Accuracy:
All Sensors Reporting

Diagnosis:
Replace Gas Cap

Expected Cost:
$25.00



XML Made Simple

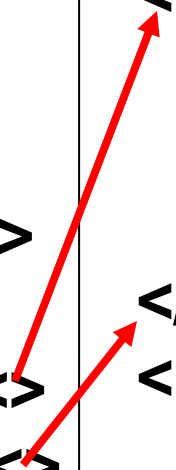
XCCDF - eXtensible Car Care Description Format

```
<Car>
  <Description>
    <Year> 1997 </Year>
    <Make> Ford </Make>
    <Model> Contour </Model>
  <Maintenance>
    <Check1> Gas Cap = On <>
    <Check2>Oil Level = Full <>
  </Maintenance>
 </Description>
</Car>
```

OVAL – Open Vehicle Assessment Language

```
<Checks>
  <Check1>
    <Location> Side of Car <>

    <Procedure> Turn <>
  </Check1>
  <Check2>
    <Location> Hood <>
    </Procedure> … <>
  </Check2>
</Checks>
```



FISMA Compliance

Starting Point

FIPS 199 / SP 800-60

Security Categorization

Defines category of information system according to potential impact of loss

FIPS 200 / SP 800-53

Security Control Selection

Selects minimum security controls (i.e., safeguards and countermeasures) planned or in place to protect the information system

SP 800-37

Security Control Monitoring

Continuously tracks changes to the information system that may affect security controls and assesses control effectiveness

SP 800-53 / FIPS 200 / SP 800-30

Security Control Refinement

Uses risk assessment to adjust minimum control set based on local conditions, required threat coverage, and specific agency requirements

SP 800-37

System Authorization

Determines risk to agency operations, agency assets, or individuals and, if acceptable, authorizes information system processing

SP 800-18

Security Control Documentation

In system security plan, provides a an overview of the security requirements for the information system and documents the security controls planned or in place

SP 800-70

Security Control Implementation

Implements security controls in new or legacy information systems; implements security configuration checklists

SP 800-53A / SP 800-26 / SP 800-37

Security Control Assessment

Determines extent to which the security controls are implemented correctly, operating as intended, and producing desired outcome with respect to meeting security requirements

Common FISMA Statements

While FISMA compliance is **Important**, it can be **Complex** and **Laborious**.

"Can parts of FISMA compliance be streamlined and automated"?

"My organization spends more money on compliance than remediation".

Fundamental FISMA Questions

What are the NIST Technical Security Controls?

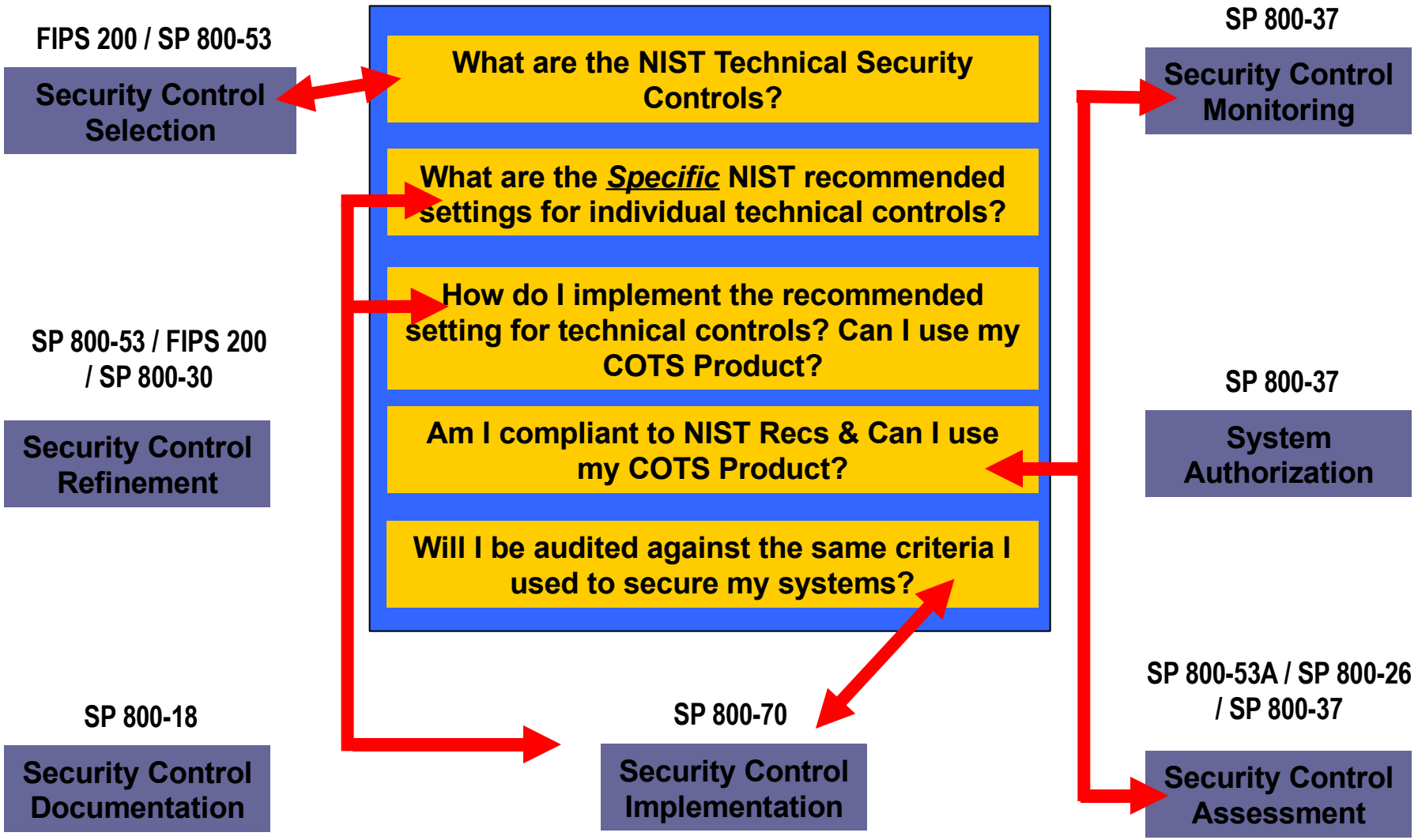
What are the *Specific* NIST recommended settings for individual technical controls?

How do I implement the recommended setting for technical controls? Can I use my COTS Product?

Am I compliant to NIST Recs & Can I use my COTS Product?

Will I be audited against the same criteria I used to secure my systems?

FISMA Documents



Automation of FISMA Technical Controls

What are the NIST Technical Security Controls?

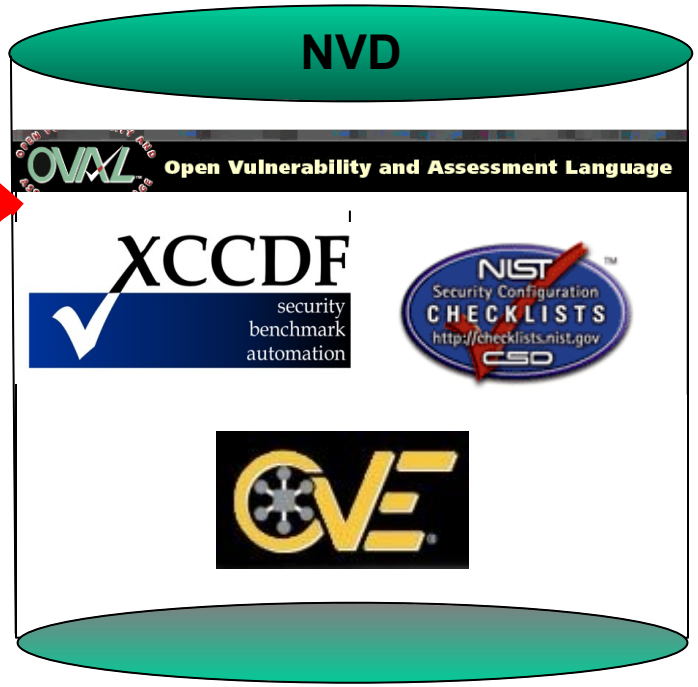
What are the *Specific* NIST recommended settings for individual technical controls?

How do I implement the recommended setting for technical controls? Can I use my COTS Product?

Am I compliant to NIST Recs & Can I use my COTS Product?

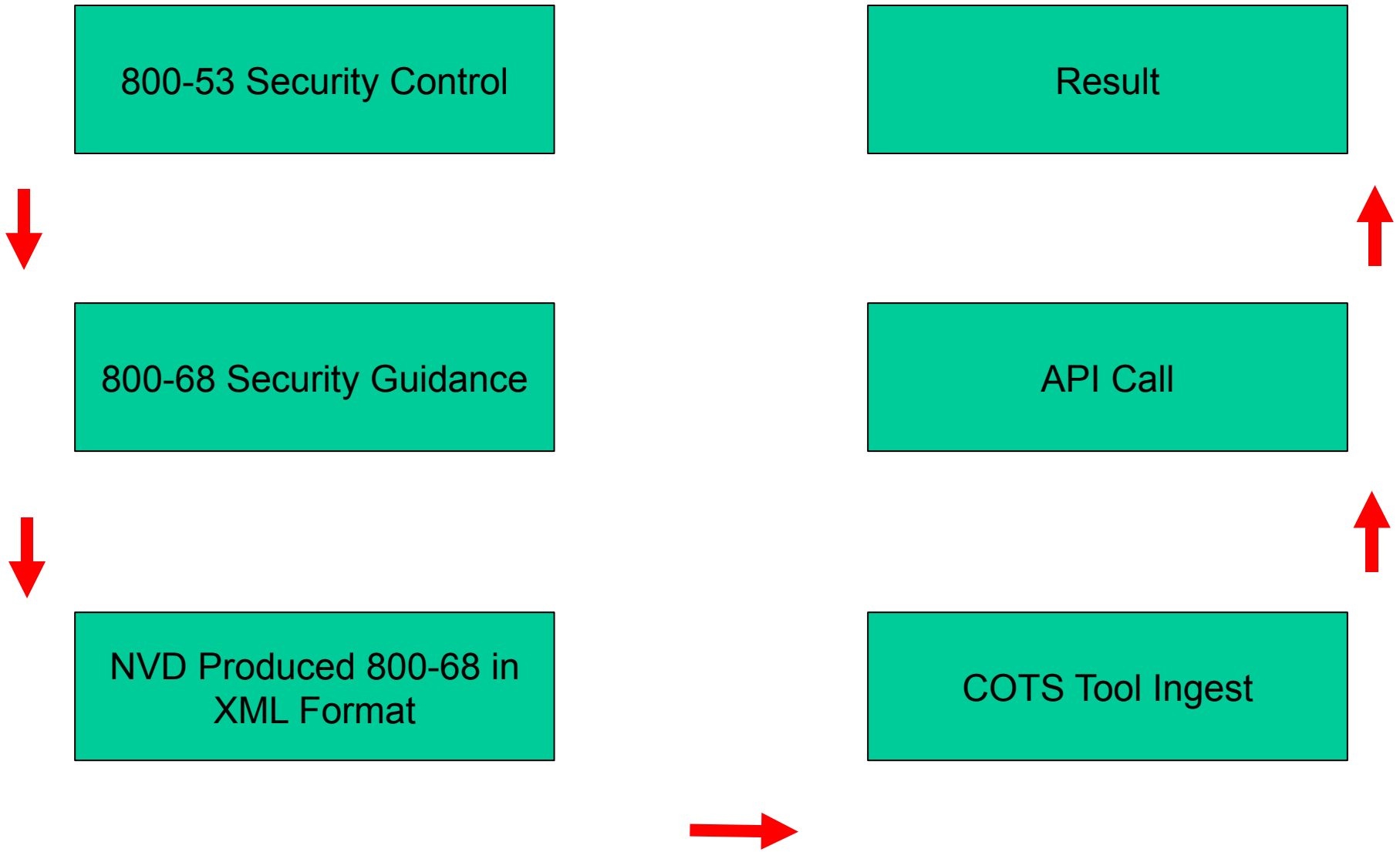
Will I be audited against the same criteria I used to secure my systems?

COTS Tools



Automated Compliance

The Connected Path



Automated Compliance

800-53 Security Control

AC-7 Unsuccessful Login

800-68 Security Guidance

AC-7: Account Lockout Duration

AC-7: Account Lockout Threshold

NVD Produced 800-68 in XML Format

```
- <registry_test id="wrt-9999" comment="Account Lockout
Duration Set to 5" check="at least 5">
- <object>
  <hive>HKEY_LOCAL_MACHINE</hive>
  <key>Software\Microsoft\Windows</key>
  <name>AccountLockoutDuration</name>
  </object>
- <data operation="AND">
  <value operator="greater than">5*</value>
```

COTS Tool Ingest

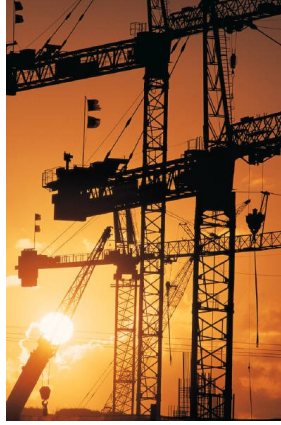
API Call

lpHKey = "HKEY_LOCAL_MACHINE"
Path = "Software\Microsoft\Windows\"
Value = "5"
sKey = "AccountLockoutDuration"
Op = ">"

Result

```
RegQueryValue (lpHKey, path, value, sKey, Value, Op);
If (Op == '>" )
if ((sKey < Value )
return (1); else
return (0);
```

Wrapping together existing initiatives



- Mitre - Common Vulnerability Enumeration (CVE)
- Mitre - Open Vulnerability Assessment Language (OVAL)
- NSA - eXtensible Configuration Checklist Description Format (XCCDF)
- FIRST – Common Vulnerability Scoring System (CVSS)
- NIST - National Vulnerability Database
- NIST - NIST Checklist Program

Existing NIST Products

National Vulnerability Database

- 2.2 million hits per month
- 20 new vulnerabilities per day
- Integrated standards:
  - CVE — 244 products
  - CVSS — 20 vendors
  - OVAL — 8 vendors, 24 products

Checklist Program

- 91 separate guidance documents
- Covers 140 IT products

National Vulnerability Database

NVD is a comprehensive cyber security vulnerability database that:

- Integrates all publicly available U.S. Government vulnerability resources
- Provides references to industry resources.
- It is based on and synchronized with the CVE vulnerability naming standard.
- XML feed for all CVEs

http://nvd.nist.gov

Sponsored by DHS National Cyber Security Division/US-CERT

NIST National Institute of Standards and Technology

National Vulnerability Database

a comprehensive cyber vulnerability resource

Welcome to NVD!!

NVD is a comprehensive cyber security vulnerability database that integrates all publicly available U.S. Government vulnerability resources and provides references to industry resources. It is based on and synchronized with the CVE vulnerability naming standard.

Resource Status

NVD contains:
16418 CVE Vulnerabilities
54 US-CERT Alerts
1245 US-CERT Vuln Notes
1162 Oval Queries
Last updated:
04/14/06
Publication rate:
17 vulnerabilities / day

Workload Index

Vulnerability Workload Index: 6.89

Email List

Enter your e-mail address and press "Add" to receive NVD announcements.

Add

About Us

NVD is a product of the

Search CVE Vulnerability Database

(Perform Advanced Search)

Keyword search: Search All

Try a product or vendor name
Try a CVE standard vulnerability name or OVAL query
Only vulnerabilities that match ALL keywords will be returned
Linux kernel vulnerabilities are categorized separately from vulnerabilities in specific Linux distributions

Search last 3 months | Search last 3 years

Show only vulnerabilities that have the following associated resources:

- ☐ US-CERT Technical Alerts
- ☐ US-CERT Vulnerability Notes
- ☐ OVAL Queries

Recent CVE Vulnerabilities

CVE-2006-1790 **Publish Date:** 4/14/2006
A regression fix in Mozilla Firefox 1.0.7 allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary code via the InstallTrigger.install method, which leads to memory corruption.

CVE-2006-1738 **Publish Date:** 4/14/2006
Unspecified vulnerability in Mozilla Firefox and Thunderbird 1.x before 1.5 and 1.0.x before 1.0.8, Mozilla Suite before 1.7.13, and SeaMonkey before 1.0 allows remote attackers to cause a denial of service (crash) by changing the (1) -moz-grid and (2) -moz-grid-group display styles.

CVE-2006-1737 **Publish Date:** 4/14/2006
Integer overflow in Mozilla Firefox and Thunderbird 1.x before 1.5 and 1.0.x before 1.0.8, Mozilla Suite before 1.7.13, and SeaMonkey before 1.0 allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary bytecode via JavaScript with a large regular expression.

CVE-2006-1742 (Firefox, Thunderbird, Mozilla suite, SeaMonkey)
Publish Date: 4/14/2006 **CVSS Severity:** 2.3 (Low)
The JavaScript engine in Mozilla Firefox and Thunderbird 1.x before 1.5 and 1.0.x before 1.0.8, Mozilla Suite before 1.7.13, and SeaMonkey before 1.0 does not properly handle temporary variables that are not garbage collected, which might allow remote attackers to trigger operations on freed memory and cause memory corruption.

CVE-2006-1741 (Firefox, Thunderbird, Mozilla suite, SeaMonkey)

Sponsored by DHS National Cyber Security Division/US-CERT

NIST National Institute of Standards and Technology

National Vulnerability Database

a comprehensive cyber vulnerability resource

Welcome to NVD!!

NVD is a comprehensive cyber security vulnerability database that integrates all publicly available U.S. Government vulnerability resources and provides references to industry resources. It is based on and synchronized with the CVE vulnerability naming standard.

Resource Status

NVD contains:
16418 CVE Vulnerabilities
54 US-CERT Alerts
1245 US-CERT Vuln Notes
1162 Oval Queries
Last updated:
04/14/06
Publication rate:
17 vulnerabilities / day

Workload Index

Vulnerability Workload Index: 6.89

Email List

Enter your e-mail address and press "Add" to receive NVD announcements.

Add

About Us

NVD is a product of the

There are **28** matching records. Displaying matches **1** through **20**.

Next 20 Matches

CVE-2006-0012 TA06-101A VU#641460

Summary: Unspecified vulnerability in Windows Explorer in Microsoft Windows 2000 SP4, XP SP1 and SP2, and Server 2003 SP1 allows remote attackers to execute arbitrary code via attack vectors involving COM objects and "crafted files and directories," aka the "Windows Shell Vulnerability."
Published: 4/11/2006
CVSS Severity: 5.6 (Medium)

CVE-2006-0003 TA06-101A VU#234812

Summary: Unspecified vulnerability in the RDS.Dataspace ActiveX control, which is contained in ActiveX Data Objects (ADO) and distributed in Microsoft Data Access Components (MDAC) 2.7 and 2.8, allows remote attackers to execute arbitrary code via unknown attack vectors.
Published: 4/11/2006
CVSS Severity: 5.6 (Medium)

CVE-2006-1189 TA06-101A VU#341028

Summary: Unspecified vulnerability in Microsoft Internet Explorer 5.01 through 6 allows remote attackers to execute arbitrary code via a crafted URL with double-byte characters, aka the "Double Byte Character Parsing Memory Corruption Vulnerability."
Published: 4/11/2006
CVSS Severity: 10.0 (High)

CVE-2006-1188 TA06-101A VU#824324

Summary: Microsoft Internet Explorer 5.01 through 6 allows remote attackers to execute arbitrary code via HTML elements with a certain crafted tag, which leads to memory corruption.
Published: 4/11/2006
CVSS Severity: 7.0 (High)

CVE-2006-1186 TA06-101A

Summary: Microsoft Internet Explorer 5.01 through 6 allows remote attackers to execute arbitrary code via by instantiating the (1) Mdt2gddr.dll, (2) Mdt2dd.dll, and (3) Mdt2gddo.dll COM objects as ActiveX controls, which leads to memory corruption.
Published: 4/11/2006
CVSS Severity: 10.0 (High)

CVE-2006-1185 TA06-101A VU#503124

Summary: Unspecified vulnerability in Microsoft Internet Explorer 5.01 through 6 allows

Example Command line

- C:\>ovaldi.exe \
- –o windows.definitions.xml \
- -d payne\d \
- -m \
- -r payne\r \
- -p > payne\1 \
- 2> payne\2

- MITRE's demo interpreter
- -o OVAL file to use
- -d data – systems characteristics file
- -m DO NOT check md5 sum
- -r results file
- -p print much info

Trying just OVAL id 1020

- OVAL-ID:
  oval:org.mitre.oval:def:1020

Offline OVAL analysis

- "-i – Specifies the pathname of a System Characteristics file that is to be used as the basis of the analysis. In this mode, the Interpreter does not perform data collection on the local system, but relies upon the input file, which may have been generated on another system"

Future plans for OVALdi development:

- We are currently committed to implementing a number of new probes by early September. Let me make sure we do not duplicate any efforts here. here is the list that I plan to start working on in the next few weeks:

  win-def:passwordpolicy_test
  win-def:lockoutpolicy_test
  win-def:auditeventpolicy_test
  win-def:file effective rights_test
  win-def:sid_test

Variable Support

- MITRE will also implement external variables soon and the remaining function types defined for use with local_variables:

  oval-def:BeginFunctionType
  oval-def:EndFunctionType
  oval-def:EscapeRegexFunctionType
  oval-def:SplitFunctionType
  oval-def:EscapeRegexFunctionType

References

Slides taken from:

1. "OVAL Tutorial". John Baker, MITRE. July 12th 2006.

2. "Automated Security Measurement FISMA Technical Control Automation". Stephen Quinn & Peter Mell, NIST.

Both sets of slides available at:

http://oval.mitre.org/oval/documents/docs-06/oval_developer_days_slides.zip

or

http://tinyurl.com/k9oez

Your Key to Security

Tools Used

- XMLSpy - www.altova.com
- Eclipse WTP - http://www.eclipse.org/webtools/
- XPath Explorer - http://www.purpletech.com/xpe/index.jsp
- Microsoft's XMLPad
  - http://tinyurl.com/ot2eo

XCCDF vs. OVAL

XCCDF

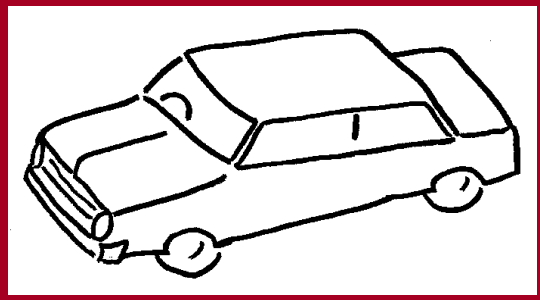
- ***High level language which defines security policies relevent to a sytem***
- ***Describes what makes up a secure system***
- ***Does <u>not</u> include info about how to assess specific systems***
- ***References OVAL tests for that platform***

OVAL

- Low level language which compares a system's state to a policy
- Performs specific tests for a specific platform
- Provides simple pass/fail results for each test

OVAL + XCCDF

Example: Car Care



XCCDF

```
<car>
 <description>
  <year>1997</year>
  <make>Ford</make>
  <model>Contour</model>
 </description>
 <maintainance>
  <check1>Gas Cap =On</>
  <check2>Oil Level =Full</>
 </maintainance>
</car>
```

OVAL

```
<checks>
  <check1>
    <location>side of car</>
    <procedure>turn</>
  </check1>
  <check2>
    <location>hood</>
    <procedure>dipstick</>
  </check2>
</checks>
```

What is OVAL?

- *Open Vulnerability Assessment Language*
- *Created by MITRE*
- *Three Components:*
  - *Language*
  - *Repository*
  - *Interpreter*

Language
http://oval.mitre.org

A collection of XML schema for representing system information, expressing specific machine states, and reporting the results of an assessment.

Language
http://oval.mitre.org

Three main schemas:

- ***System Characteristics Schema***
- ***Definition Schema***
- ***Results Schema***

OVAL definitions use the schemas to keep definitions consistent and standardized for each supported platform.

Repository
http://oval.mitre.org

The central meeting place for the OVAL Community to discuss, analyze, store, and disseminate OVAL Definitions.

Updated regularly, data files may be used in the OVAL Interpreter as well as incorporated by vendors into their v ulnerability assessment and other information security products and services.

Interpreter

http://oval.mitre.org

Proof of Concept

Security Configuration Scanner

OVAL's reference interpreter shows how: information can be collected from a computer; definitions can be used to test the system for computer vulnerabilities, configuration issues, programs, and patches; and results of the tests can be presented.

Getting Started

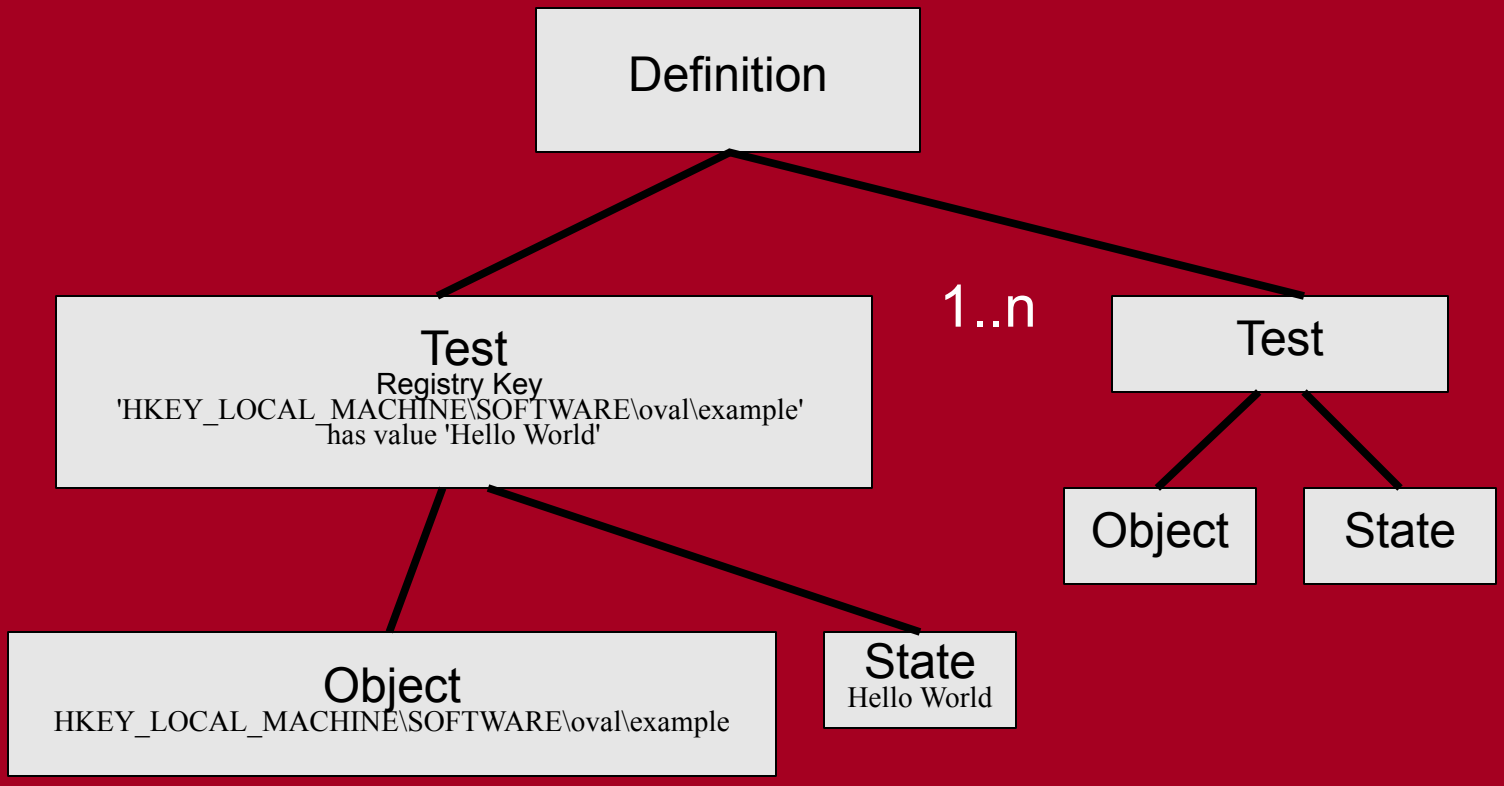
The OVAL 5 Interpreter source and binary distributions are available from http://oval.mitre.org

Download page: http://tinyurl.com/faj9g

Interpreter

- *The OVAL interpreter uses 'definitions' as input.*
- *OVAL input is expressed in XML.*
- *Each input file contains one or more definitions.*
- *Sets of pre-defined definitions exist for several operating systems/applcations.*

Definition Structure - Hello World



Object Code

```
<registry_object
id="oval:org.nucia.oval.tutorial:obj:1">
    <hive>HKEY_LOCAL_MACHINE</hive>
    <key>SOFTWARE\oval</key>
    <name>example</name>
</registry_object>
```

State Code

```
<registry_state
id="oval:org.nucia.oval.tutorial:ste:1">
    <value operation="equals">Hello World</value>
</registry_state>
```

Test Code

```
<registry_test id="oval:org.nucia.oval.tutorial:tst:1"
check="all">
    <object object_ref="oval:org.nucia.oval.tutorial:obj:1" />
    <state state_ref="oval:org.nucia.oval.tutorial:ste:1" />
</registry_test>
```

- The OVAL Interpreter currently supports the following tests:

1. ind-def:environmentvariable_test
2. ind-def:filemd5_test
3. ind-def:variable_test
4. ind-def:xmlfilecontent_test
5. win-def:file_test
6. win-def:registry_test
7. win-def:wmi_test
8. unix-def:file_test
9. unix-def:uname_test
10. unix-def:process_test
11. linux-def:inetlisteningservers_test
12. ind-def:family_test
13. linux-def:rpminfo_test

Of the following 13 tests,

Test Frequency

- textfilecontent_test happens 22 times
  process_test happens 76 times
  unknown_test happens 20 times
  family_test happens 4 times
  package_test happens 97 times
  file_test happens 3263 times
  rpmdatabase_test happens 32 times
  inetd_test happens 29 times
  swlist_test happens 92 times
  rpminfo_test happens 621 times
  registry_test happens 5043 times
  uname_test happens 699 times
  inetlisteningservers_test happens 49 times
  patch_test happens 438 times
  For a total of 10485 tests. Scanned 1706 .xml files.

Definition Code

```
<definition id="oval:org.nucia.oval.tutorial:def:1">
    <metadata>
        <title>Hello World Example</title>
        <description>
            This definition is used to introduce the OVAL
            language to those interested in writing OVAL content.
        </description>
    </metadata>
    <criteria>
        <criterion
test_ref="oval:org.nucia.oval.tutorial:tst:1"
comment="the value of the registry key equals Hello World" />
    </criteria>
</definition>
```

Definitions Document Structure

- Namespace Stuff
- Generator
- Definitions
- Tests
- Objects
- States
- Variables
- Digital Signature

Generator Code

This code provides information about the OVAL Document .

```
<generator>
    <oval:schema_version>5.0</oval:schema_version>
    <oval:timestamp>2005-10-12T18:13:45</oval:timestamp>
</generator>
```

Variables

- Define values to be obtained at run time.
- Represent an array of values.
- Three Types:
- local_variable
- external_variable
- constant_variable

```
<constant_variable datatype=”string” id=”var1”>
    <value>\system32</value>
</constant_variable>
```

Running

- *Open a command prompt.*
- *~\Program Files\OVAL\ovaldi.exe -m -o hello_world.xml -d data_file.xml -r results_file.xml"*

This will invoke the interpreter to run the hello world definition (-o), ignoring the MD5 of the definitions file (-m). It will write the raw data to the data_file (-d) and the results to the results_file (-r).

Results

The interpreter generates a results XML document.

The results document contains:

- *Generic system information*
- *The original definitions document*
- *A true/false result for each definition*

Your Key to Security

So What?

What's this good for?

- ***XCCDF and other high-level tools***
- ***Java Web-Apps***
- ***…***

CIS XCCDF Benchmark

- *Only currently-available XCCDF benchmark tool*
- *Uses only proprietary OVAL definitions written by CIS*
- *Commercial-ware*
- *Demonstrates the ability to customize OVAL for specific needs.*
- *Uses OVAL version 4 which encapsulates objects and states in the corresponding tests.*

Example XCCDF

```
...
<Rule id='restrict-guest-acc-rul" selected='true">
    <status date="2005-01-30">draft</status>
        <title>Restrict Guest Access</title>
        <description>
            <htm:p>For more information on this setting, please review the description for item 2.2.4.</htm:p>
        </description>
        <check system="http://oval.mitre.org/XMLSchema/oval">
        <check-export export-name="var-0006" v alue-id="restrict-guest-acc-val"></check-export>
        <check-content-ref href='cis-winxp-oval.xml" name="OVAL10005"></check-content-ref>
        </check>
</Rule>
...
```

Corresponding OVAL

```
<definition class='compliance" id='OVAL10005">
  …
  <criteria>
    <configuration operation='AND ">
      <criterion comment='" test_ref='cis-0006"></criterion>
    </configuration>
  </criteria>
</definition>
```

Corresponding OVAL

```
<registry_test
xmlns="http://oval.mitre.org/XMLSchema/oval#windows"
check="all" comment="Restrict Guest Access" id="cis-0006">
...
  <object>
    <hive>HKEY_LOCAL_MACHINE</hive>
    <key>SYSTEM\CurrentControlSet\
    Services\Eventlog\Security</key>
    <name>RestrictGuestAccess</name>
  </object>
  <data operation="AND">
    <type>reg_dword</type>
    <value datatype="binary" operator="equals"
    v ar_ref="var-0006"></value>
  </data>
</registry_test>
```

More OVAL Examples

Definition oval:org.mitre.oval.org:def:105

OVAL-ID: oval:org.mitre.oval:def:105		Date: 2006-06-26
Status:	ACCEPTED	**Description:**
Class:	inventory	The operating system installed on the system is Microsoft Windows XP.
Ref-ID:		
Schema Version:	5	
Platform(s):	Microsoft Windows XP	

Definition Synopsis:

- the installed operating system is part of the Microsoft Windows family
- AND Windows XP is installed

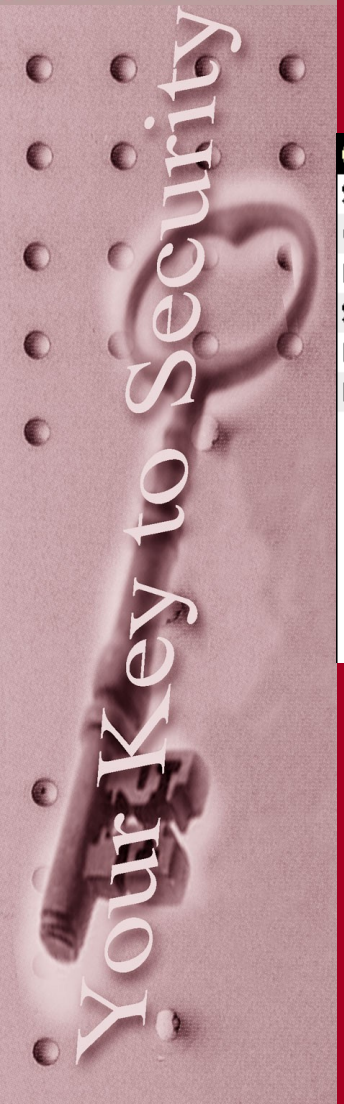
definition-105.xml

Run_it.bat

output.log

raw_data.xml

results.xml



More OVAL Examples

Definition oval:org.mitre.oval.org:def:100011

OVAL-ID: oval:org.mitre.oval:def:100011		Date: 2006-07-03
Status:	INTERIM	**Description:**
Class:	vulnerability	Firefox 1.0.3 and 1.0.4, and Netscape 8.0.2, allows remote attackers to execute arbitrary code by tricking the user into using the "Set As Wallpaper" (in Firefox) or "Set as Background" (in Netscape) context menu on an image URL that is really a javascript: URL with an eval statement, aka "Firewalling."
Ref-ID:	CVE-2005-2262	
Schema Version:	5	
Platform(s):	Microsoft Windows 2000	

Definition Synopsis:

- Mozilla Firefox version 1.0.4 or earlier is installed
  - Firefox version 1.0.4 or earlier is installed
  - AND Mozilla Firefox version 1.0.4 or earlier is installed
- AND NOT Mozilla Firefox version 1.0.2 or earlier is installed
  - Firefox version 1.0.2 or earlier is installed
  - AND Mozilla Firefox version 1.0.2 or earlier is installed

definition-100011.xml

Run_it.bat

output.log

Raw_data.xml

Results.xml

Your Key to Security

Future Work

1. Write OVAL definitions.
2. Build an open source XCCDF editor.
3. Build an open source XCCDF interpreter.
4. Write XCCDF/OVAL for securing JAVA web applications.

Apache specific tests

- version_test
  - The version test is used to check the version of Apache installed system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an apache_object and the optional state element specifies the data to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType.

Independent Definition

- family_test
  - >The family_test element is used to check the family a certain system belongs to. This test basically allows the high level system types (window, unix, ios, etc.) to be tested. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a family_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType.

file_md5_test

- "The file_md5 test is used to check the md5 associated with a specified file. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a file_md5_object and the optional state element specifies the md5 to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

Linux: dpkginfo_test

- The dpkginfo test is used to check information for a given DPKG package. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a dpkginfo_object and the optional state element specifies the data to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType.

inetlistening servers_test

- "The inet listening servers test is used to check What applications are listening on the network. It is generally using the parsed output of running the command netstat -tuwlnpe with root privilege. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetlistening servers_object and the optional state element specifies the data to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

rpminfo_test

- "The rpm info test is used to check the RPM header information for a given RPM package. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a rpminfo_object and the optional state element specifies the data to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

OS X: accountinfo_test

- "User account information (username, uid, gid, etc.) See netinfo(5) for field information, niutil(1) for retrieving it. We may need/want to add in data elements for things like authentication_authority, generateduid, mcx_settings (restricted account settings"

inetlistenningservers_test

- "This test's purpose is generally used to check if a program is listening on the network, either for a new connections or as part of an ongoing connection. It is generally speaking the parsed output of running the command netstat -tuwlnpe with root privilege."

OS X: nvram_test

- "This test pulls data from the 'nvram -p' output."

pwpolicy_test

- "This test pulls data from the 'pwpolicy -getpolicy' output. The actual values get stored under /var/db/netinfo/local.nidb/ in a Store.# file. Is this test actually needed, or can the text file content test be used instead?"

accountinfo_test

- "User account information (username, uid, gid, etc.) See netinfo(5) for field information, niutil(1) for retrieving it. We may need/want to add in data elements for things like authentication_authority, generateduid, mcx_settings (restricted account settings)."

Solaris: isainfo_test

- "The isainfo test reveals information about the instruction set architectures. This information can be retrieved by the isainfo command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an isainfo_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

package_test

- "The package test is used to check information associated with different packages installed on the system. The information used by this test is modeled after the /usr/bin/pkginfo command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

patch_test

- "The patch test is used to check information associated with different patches installed on the system. The information being tested is based off the /usr/bin/showrev -p command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

UNIX: file_test

- *"The file test is used to check metadata associated with UNIX files, of the sort returned by either an ls command, stat command or stat() system call. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a file_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."*

inetd_test

- "The inetd test is used to check information associated with different Internet services. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

interface_test

- "The interface test enumerate various attributes about the interfaces on a system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an interface_object and the optional state element specifies the interface information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

password_test

- "The password test is used to check metadata associated with the UNIX password file, of the sort returned by the passwd command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a password_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

process_test

- "The process test is used to check information found in the UNIX processes. It is equivalent to parsing the output of the ps command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a process_object and the optional state element specifies the process information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

runlevel_test

- "The runlevel test is used to check information about which runlevel specified service are scheduled to exist at. For more information see the output generated by a chkconfig –list. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a runlevel_object and the optional state element specifies the data to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

shadow_test

- "The shadow test is used to check information from the /etc/shadow file for a specific user. This file contains a user's password, but also their password aging and lockout information. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

uname_test

- "The uname test reveals information about the hardware the machine is running on. This information is the parsed equivalent of uname -a. For example: "Linux quark 2.6.5-7.108-default #1 Wed Aug 25 13:34:40 UTC 2004 i686 i686 i386 GNU/Linux" or "Darwin TestHost 7.7.0 Darwin Kernel Version 7.7.0: Sun Nov 7 16:06:51 PST 2004; root:xnu/xnu-517.9.5.obj~1/RELEASE_PPC Power Macintosh powerpc". It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a uname_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

Windows: accesstoken_test

- "The access token test is used to check the properties of a Windows' access token as well as individual privileges and rights associated with it. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an accesstoken_object and the optional state element specifies the data to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

active directory_test

- "The active directory test is used to check information about specific entries in active directory. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an active directory_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

auditeventpolicy_test

- "The auditeventpolicy test is used to check different types of events the system should audit. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a auditeventpolicy_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

Windows: file_test

- "The file test is used to check metadata associated with Windows files. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a file_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

file_audited_permissions_test

- "The file_audit_permissions test is used to check the audit permissions associated with Windows files. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a file_audited_permissions_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

file effective rights_test

- "The file effective rights test is used to check the effective rights associated with Windows files. Note that the trustee's effective access rights are the access rights that the ACL grants to the trustee or to any groups of which the trustee is a member. The file effective rights_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a file effective rights_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

group_test

- "The group test allows the different users that belong to specific groups be tested. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a group_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

interface_test

- "The interface test enumerate various attributes about the interfaces on a system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an interface_object and the optional state element specifies the interface information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

lockoutpolicy_test

- *"The lockoutpolicy test enumerates various attributes associated with lockout information for users and global groups in the security database. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a lockoutpolicy_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."*

metabase_test

- "The metabase test is used to check information found in the Windows metabase. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a metabase_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

passwordpolicy_test

- "The password policy test is used to check specific policy associated with passwords. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a passwordpolicy_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."
- "NOTE: This information is stored in the SAM or Active Directory but is encrypted or hidden so the registry_test and activedirectory_test are of no use. If this can be figured out, then the password_policy test is not needed."

port_test

- *"The port test is used to check information about the available ports on a Windows system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a port_object and the optional state element specifies the port information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."*

process_test

- "The process test is used to check information found in the Windows processes. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a process_object and the optional state element specifies the process information to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

registry_test

- "The registry test is used to check metadata associated with Windows registry key. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a registry_object and the optional state element specifies the registry data to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

regkeyauditedpermissions_test

- "The registry key audited permissions test is used to check the audit permissions associated with Windows registry keys. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a regkeyauditedpermissions_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

regkeyeffectiverights_test

- *"The registry key effective rights test is used to check the effective rights associated with Windows files. Note that the trustee's effective access rights are the access rights that the ACL grants to the trustee or to any groups of which the trustee is a member. The regkeyeffectiverights_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a regkeyeffectiverights_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."*

sid_test

- *"The sid test is used to check properties associated with the specified sid. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a sid_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."*

user_test

- "The user test is used to check information fabout Windows users. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a user_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

volume_test

- "The volume test is used to check information about different storage volumes found on a Windows system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a volume_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

wmi_test

- "The wmi test is used to check information access by WMI. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wmi_object and the optional state element specifies the metadata to check. The evaluation of the test is guided by the check attribute that is inherited from the TestType."

Wmi – Windows management information

- Wmic – command line version

Advertisement for OVALtools.org's OVALdsl

- A DSL – Domain Specific Language makes tasks in a given domain easier….
- We will soon release a DSL tool for OVAL and XCCDF.
- Here's an example:

Hello World in OVAL

- From MITRE's **OVAL Definition Tutorial**
  - **http://tinyurl.com/k5hh2**
- XML is good for programs & programmers
- This is great Smart interoperable data
- This is not good for subject matter experts!
- Line counts:
  - 31 in XML
  - 10 in our DSL
- OVALdsl can be generated from OVAL XML
- OVAL XML can be generated from OVAL dsl

```
1.  <oval_definitions>
2.  <definitions>
3.  <definition id="oval:org.mitre.oval:obj:1">
4.   <metadata>
5.  <title>Hello World Example</title>
6.  <description> This definition is used to introduce the OVAL
    Language to individuals interested in writing OVAL Content.
    </description>
7.  </metadata>
8.  <criteria>
9.  <criterion test_ref="oval:org.mitre.oval:tst:1" comment="the
    value of the registry key equals Hello World"/>
10. </criteria>
11. </definition>
12. </definitions>
13. <tests>
14. <registry_test id="oval:org.mitre.oval:tst:1" check="all">
15. <object object_ref="oval:org.mitre.oval:obj:1"/>
16. <state state_ref="oval:org.mitre.oval:ste:1"/>
17. </registry_test>
18. </tests>
19. <objects>
20. <registry_object id="oval:org.mitre.oval:obj:1">
21. <hive>HKEY_LOCAL_MACHINE</hive>
22. <key>SOFTWARE\oval</key>
23. <name>example</name>
24. </registry_object>
25. </objects>
26. <states>
27. <registry_state id="oval:org.mitre.oval:ste:1">
28. <value>Hello World</value>
29. </registry_state>
30. </states>
31. </oval_definitions>
```

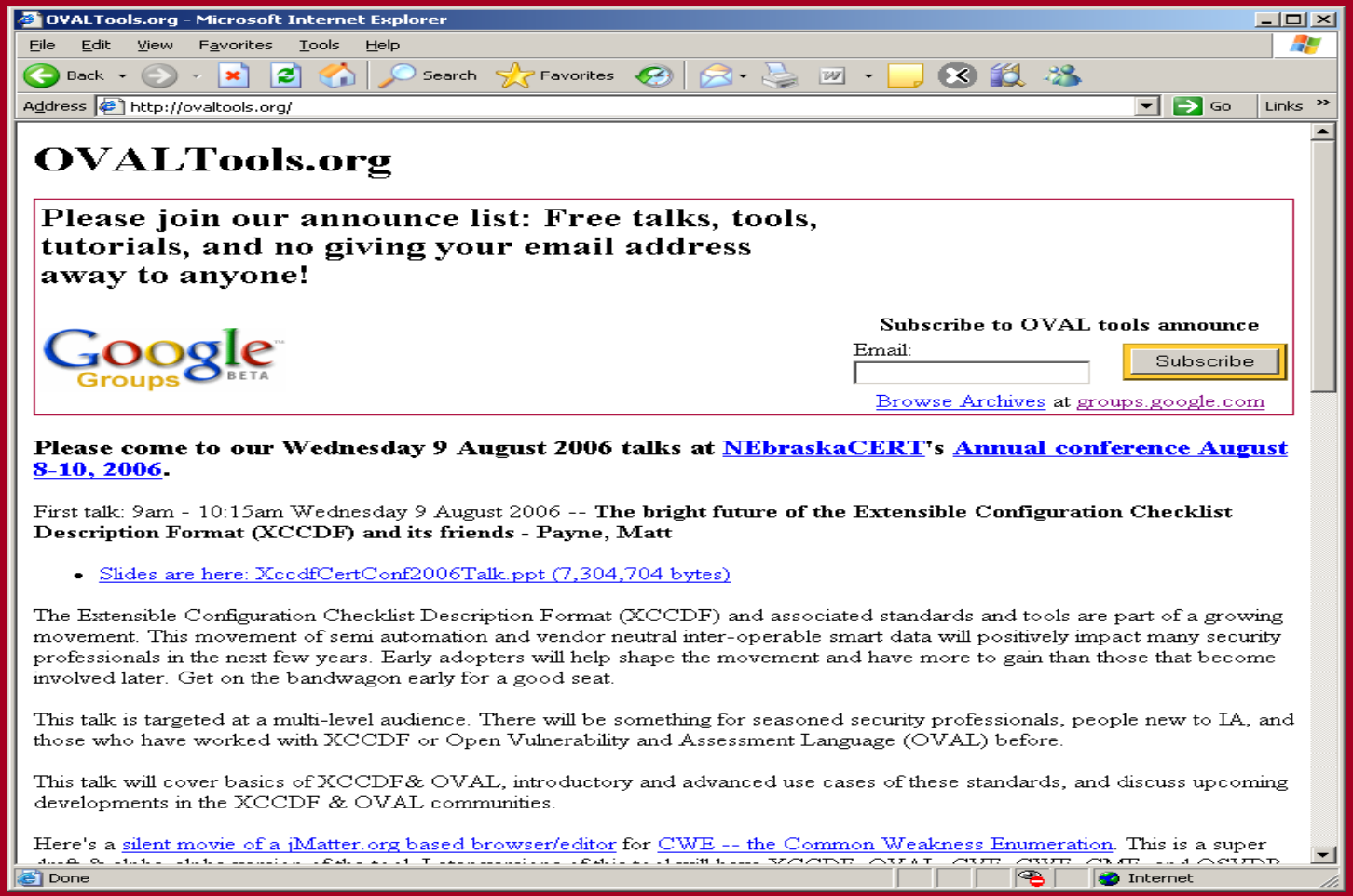
Hello World in a DSL for OVAL

```
1. def d = new Defn("vulnerability");
2. d.title = "Hello World Example";
3. d.description =
4. """
5. This definition is used to introduce the OVAL Language
6. to individuals interested in writing OVAL Content.
7. """;
8. def t= d.criteria("AND","Software Section");
9. t.comment="the value of the registry key equals Hello World";
10. t.registry_object(hive:"HKEY_LOCAL_MACHINE",key:"SOFTWARE\\oval",name:"example") == t.registry_state(value:"Hello World");
```

Blank spaces improve readabilty....

```
def d = new Defn("vulnerability");
d.title = "Hello World Example";
d.description =
"""
This definition is used to introduce the OVAL Language
to individuals interested in writing OVAL Content.
""".
def t= d.criteria("AND","Software Section");
tcomment="the value of the registry key equals Hello World";
tregistry_object(hive:"HKEY_LOCAL_MACHINE",
        key:"SOFTWARE\\oval",
        tname:"example")
        ==
        tregistry_state(value:"Hello World");
```

Advertisement for what you can do!



What's next?

- Questions? – e-mail us:
  - Payne@MattPayne.org
  - "Jason Smith": jason.theflash@gmail.com
- Signup for the new & improved OVAL tutorial
  - Add yourself to the announcement email list at ***OVALtools.org***
  - ***The next tutorial will:***
    - ***be in a lab setting at UNO***
    - ***Include use of the OVALtools OVALdsl***
    - ***And many other fun things!***