Applied SCAP Lab

Automating Security Compliance & Remediation

Shawn Wells
Office of the Chief Technologist
Director, Innovation Programs
Red Hat

Dave Smith
Infosec Engineer / Penetration Tester / OpenSCAP Upstream Maintainer
Secure Innovations, LLC

redhat.

MOTIVATION

RHEL5 STIG (U.S. Military Baseline)

- 587 compliance items
- Many are manual

Avg Time to Configure & Verify Setting	# controls	Total Time *per RHEL instance*
1 minute	* 587	9.7 hours
3 minutes	* 587	29.4 hours
5 minutes	* 587	48.9 hours

openprivacy / ansible-scap

Watch 1 | Star 2 | F

branch: master ansible-scap / provision.yml

openprivacy 14 days ago comments cleaned up

1 contributor

15 lines (12 sloc) | 0.303 kB | Raw | Blame | History

```
---

- name: All machines get OpenSCAP scanner installed
  hosts: all
  sudo: true
  roles:
    - openscap
#   - harden  -- Commented out for demo purposes only

- name: Install SCAP Security Content (SSG) and GovReady on 'dashboard'
  hosts: dashboard
  roles:
    - scap-security-guide
    - govready
```

… or a single LOC in kickstart

```
$ oscap xccdf eval \
--profile rht-ccp \
--remediate \
--report /root/scan-report.html \
/usr/share/xml/scap/content.xml
```

Compliance and Scoring

The target system did not satisfy conditions of 13 rules! Please review rule results and consider applying remediation.

Rule result breakdown

Failed rules by severity breakdown

Score

Scoring system	Score	Maximum	%
urn:xccdf:scoring:default	93.626541	100.000000	93.63%

OUR (*very ambitious*) AGENDA

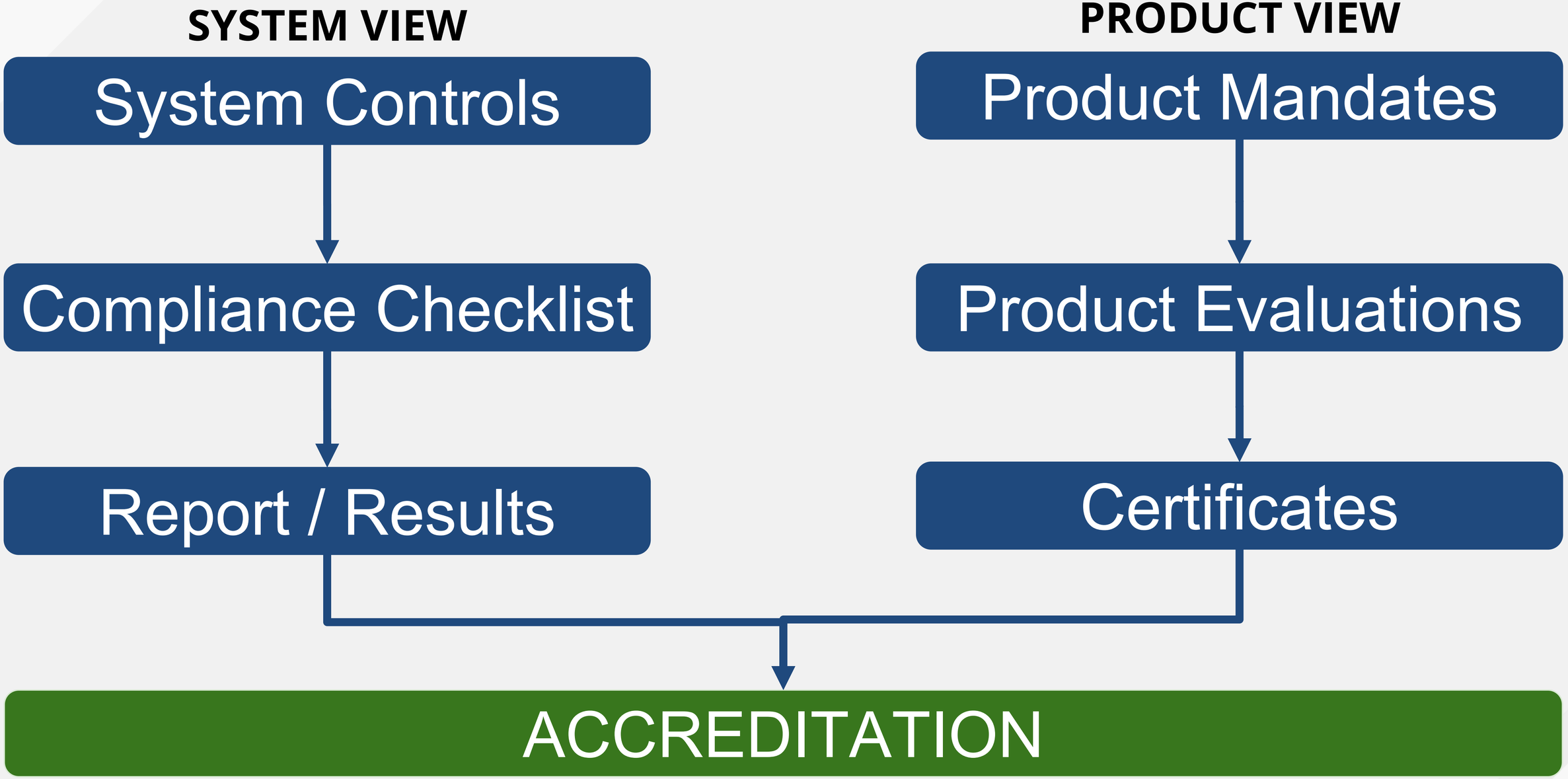
1. What's the latest in the Security Automation space?
    a. Government & Commercial Initiatives
    b. Formal and Emerging SCAP Standards

2. What tools and content are available today?
    a. For enumerating (known) software vulnerabilities
    b. For assessing configuration

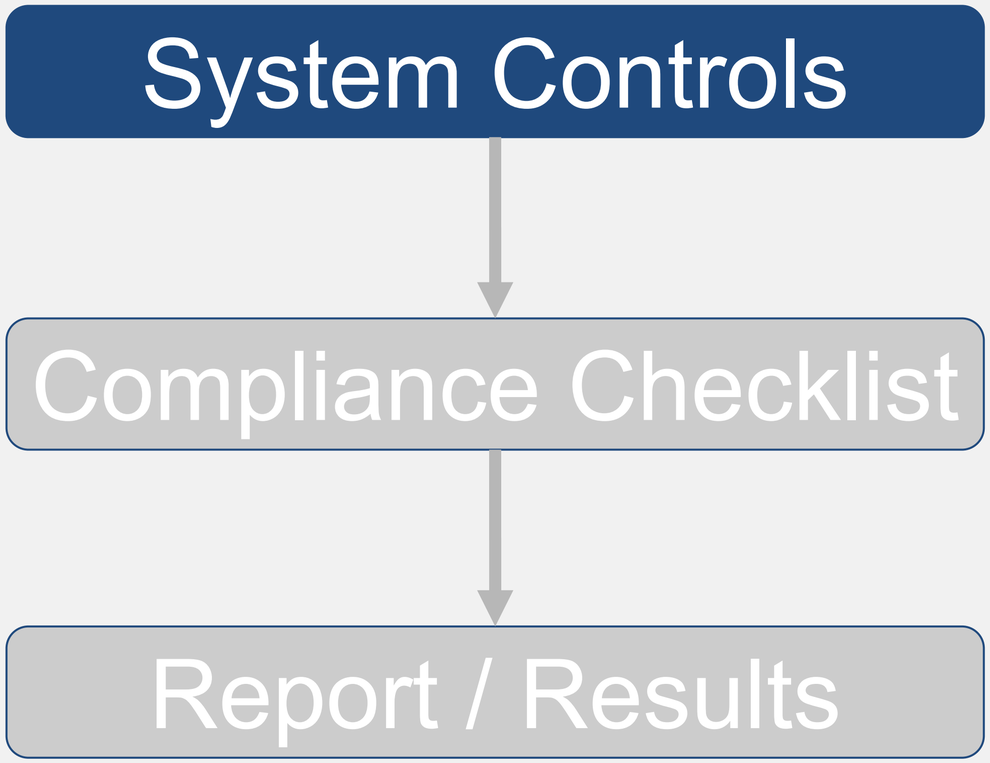
LABS

1. Install & Review SCAP profiles in RHEL 7
2. Performing a Compliance Scan
3. System Remediation
4. Creating Custom (derived) Configuration Baselines with SCAP Workbench
5. RHEL 7 “Easy Button” Installations

COMPLIANCE BIG PICTURE:
PRODUCTS AND SYSTEMS


SYSTEM VIEW
PRODUCT VIEW
System Controls
Compliance Checklist
Report / Results
Product Mandates
Product Evaluations
Certificates
ACCREDITATION

SYSTEM VIEW OF ACCREDITATION



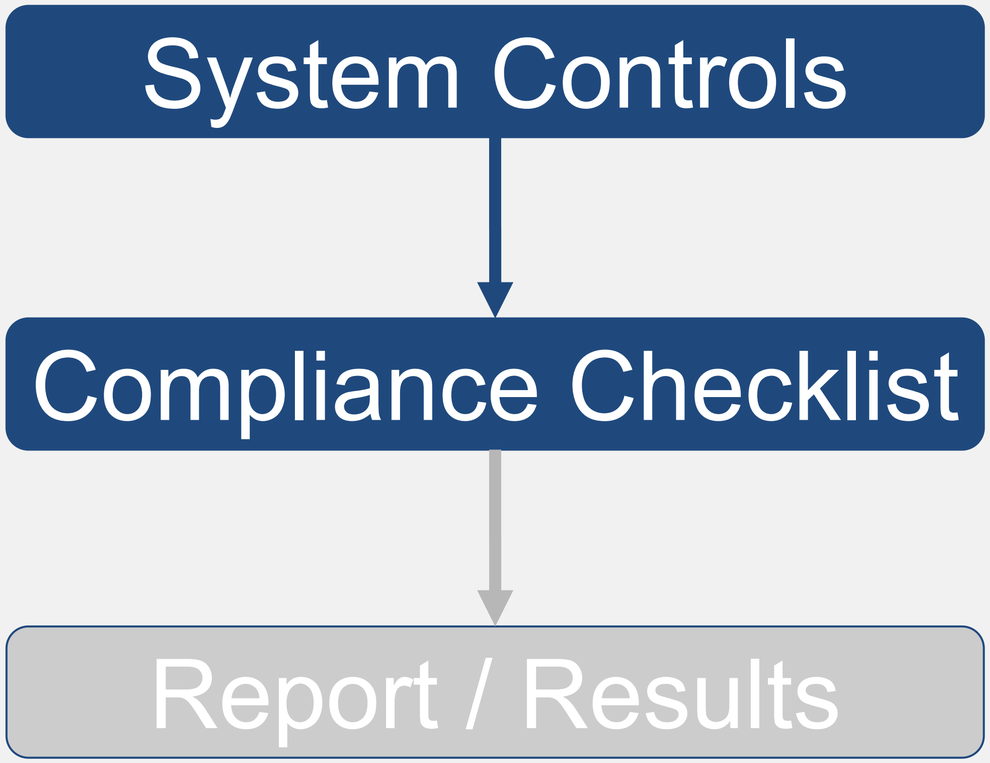
NIST 800-53

FedRAMP

CNSSI 1253

PCI

SYSTEM VIEW OF ACCREDITATION

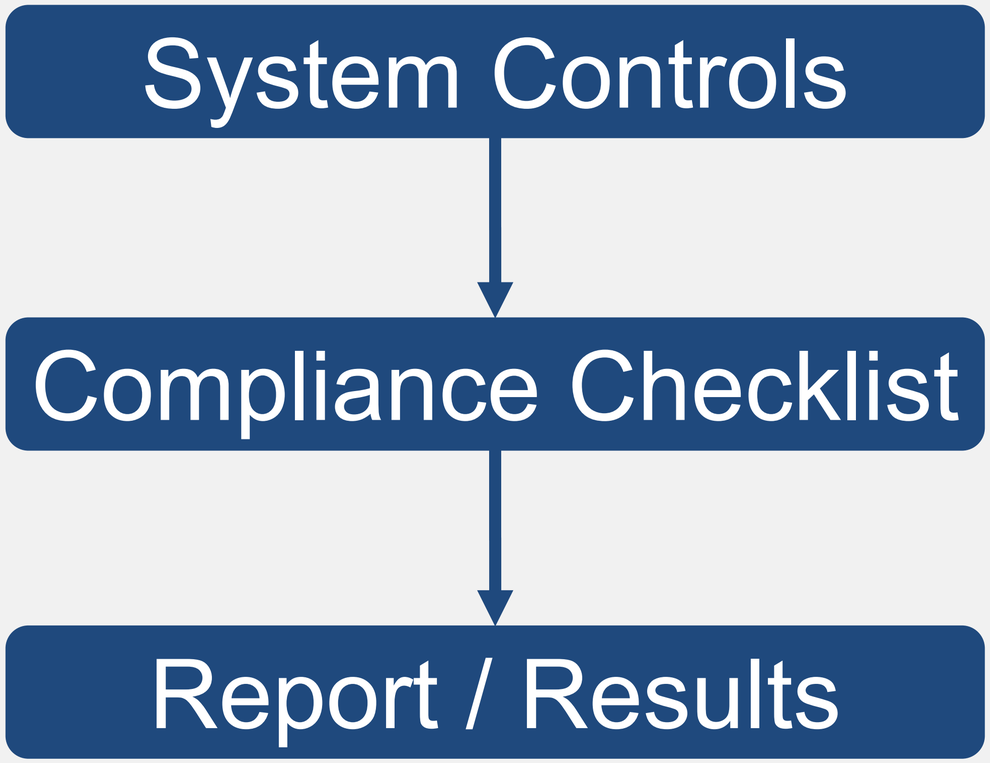


DISA STIGs

NSA SNAC Guides

CIS Benchmarks

SYSTEM VIEW OF ACCREDITATION



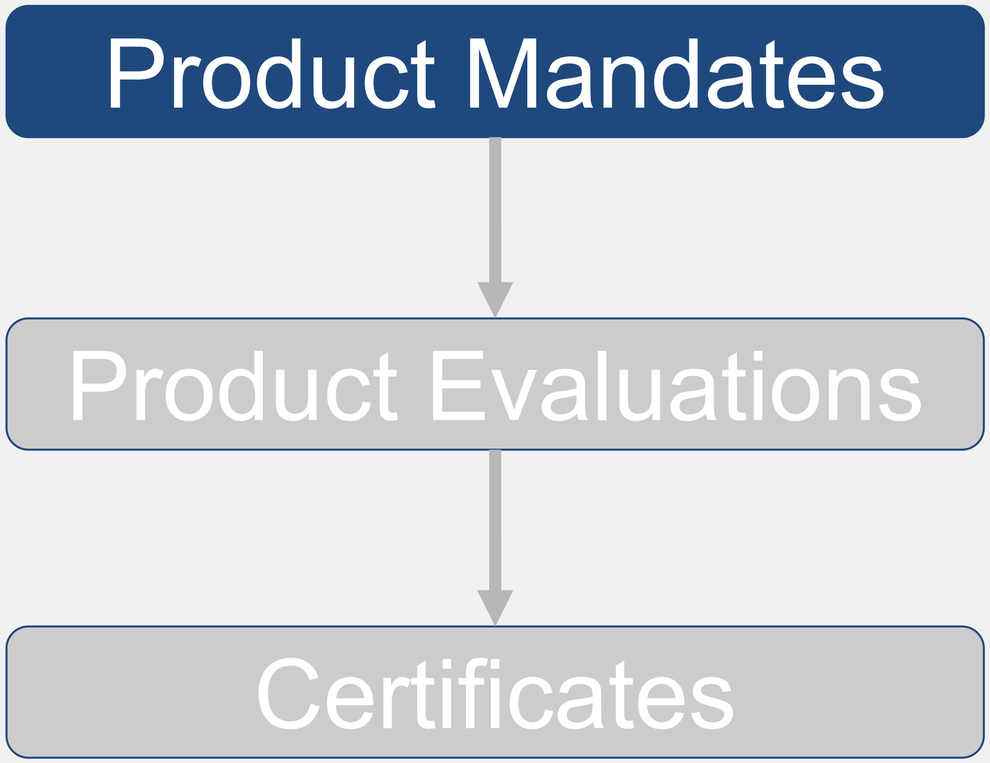
Tenable Nessus

SECSCAN

SPAWAR SCC

OpenSCAP

PRODUCT VIEW OF ACCREDITATION



Common Criteria

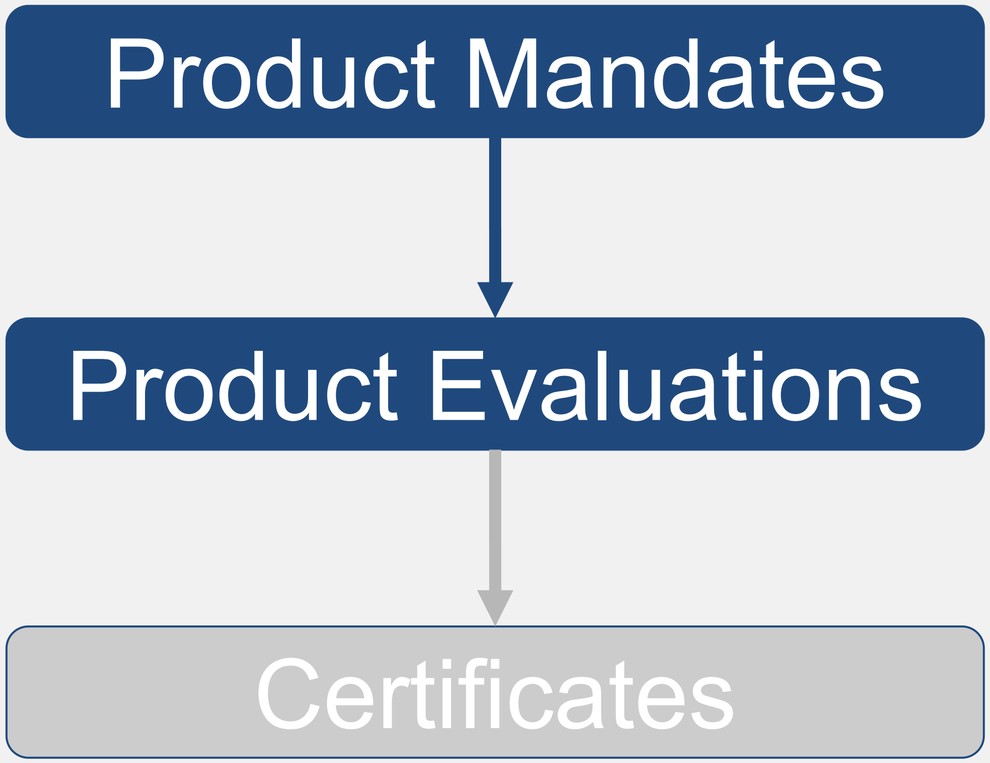
FIPS 140-2

…. wait… what's COMMON CRITERIA?

- international framework for specifying and testing security functional and assurance requirements in IT products
  - through the use of Protection Profiles (PPs)

- vendors can then implement and/or make claims about the security attributes of their products, and testing laboratories can evaluate the products to determine if they actually meet the claims.

PRODUCT VIEW OF ACCREDITATION

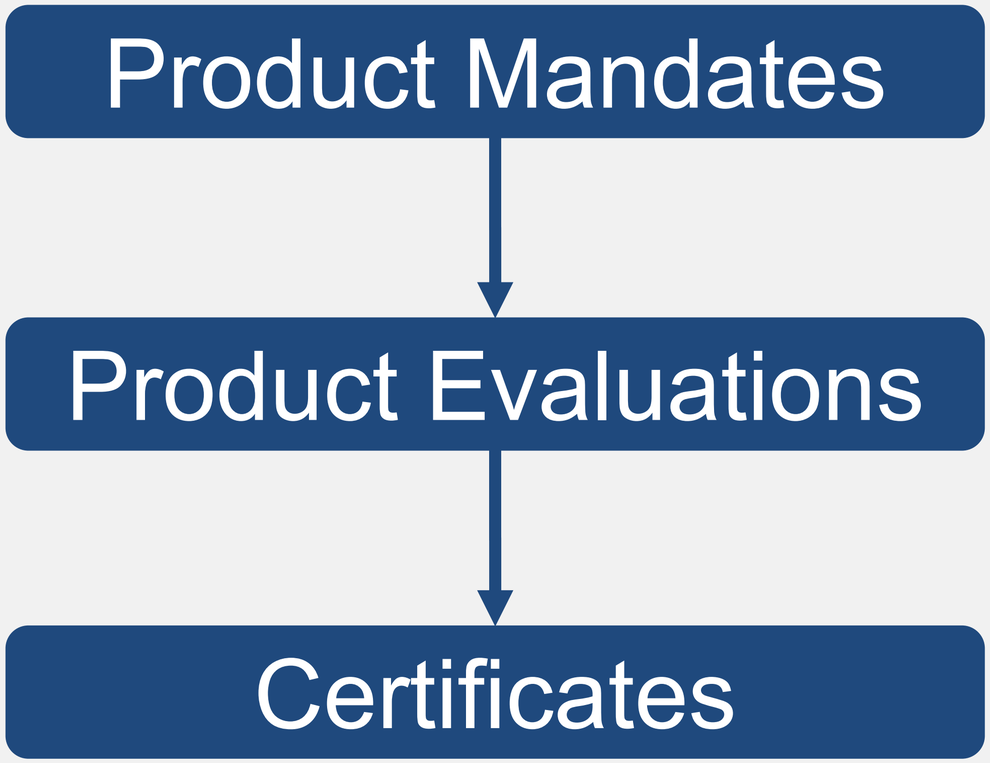


Operating System Protection Profile

Server Virtualization Protection Profile

FIPS Validation

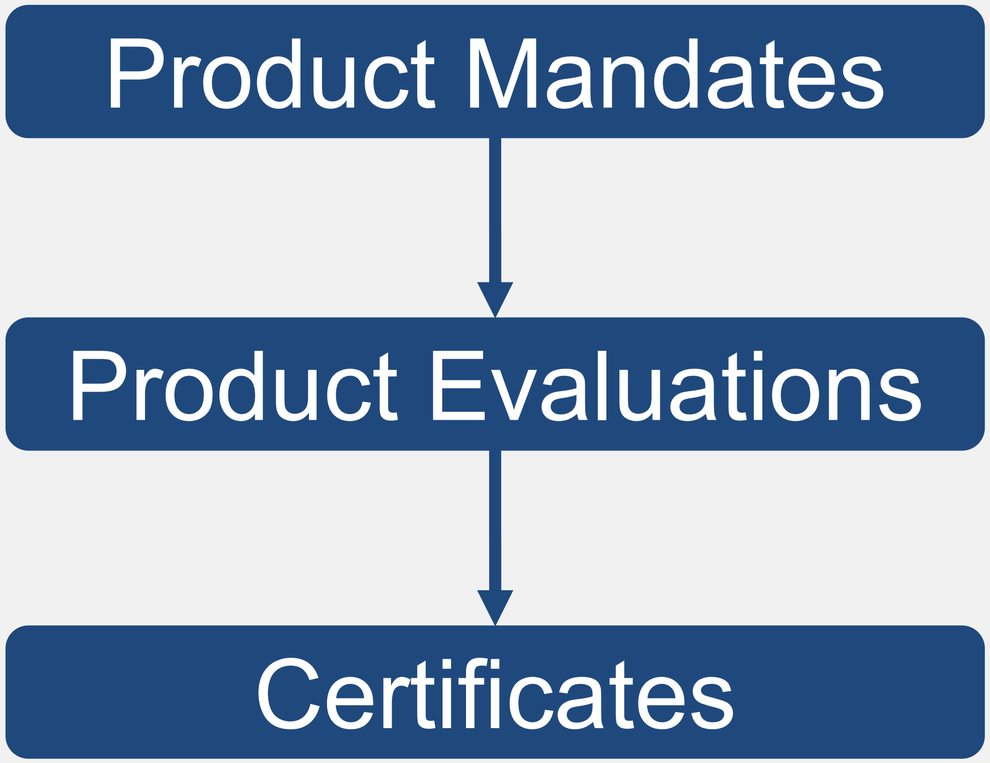
PRODUCT VIEW OF ACCREDITATION



NIAP Product Compliant List

FIPS Crypto Module Validation List

PRODUCT VIEW OF ACCREDITATION



1-2 years+

Costly ($millions)

COMMON CRITERIA - *REVAMPED*

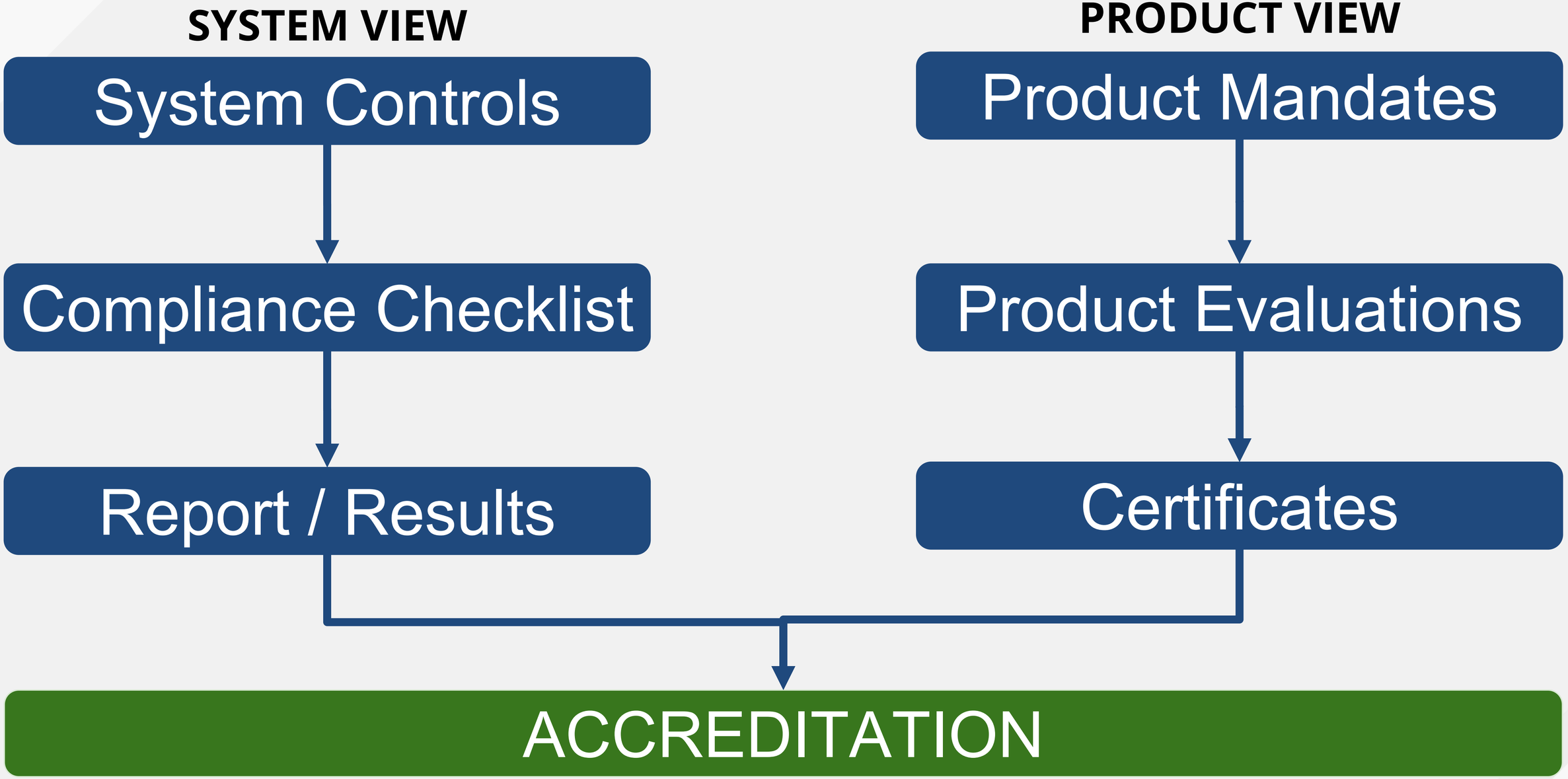
- Requirements specified in *Protection Profiles*
  - see https://www.niap-ccevs.org
  - development on https://github.com/commoncriteria
  - revamped OS Protection Profile due this July

- Dramatically reduced evaluation time and cost
  - 90 days possible, 180 max
  - compliance checklist produced during evaluation (SCAP)
  - list of system controls provided for evaluated products

COMMON CRITERIA - *REVAMPED*

- DISA STIG creation through ~25 selectable “management functions”
- DoD specific values expressed in *DoD Annexes to Protection Profiles* (succeeding SRGs)
- Remember…
  - RHEL5 STIG: 587 rules
  - RHEL6 STIG: ~255
  - RHEL.future STIG: est. < 100

Management Function
configure minimum password length
configure minimum number of special characters in password
configure minimum number of numeric characters in password
configure minimum number of uppercase characters in password
configure minimum number of lowercase characters in password
enable/disable screen lock
configure screen lock inactivity timeout
configure remote connection inactivity timeout


SYSTEM VIEW
PRODUCT VIEW
System Controls
Compliance Checklist
Report / Results
Product Mandates
Product Evaluations
Certificates
ACCREDITATION

OPEN SOURCE CONFRONTS THE C&A CHALLENGE:
SYSTEM COMPLIANCE

OpenSCAP

Community created *portfolio* of tools and content
to assess systems for known vulnerabilities.

https://github.com/OpenSCAP

2008 First commit to OpenSCAP, execution capability for SCAP on Linux

```
commit 768d2d13c7b95736738ce2a48db7f2e528c161fe
Author: Peter Vrabec <pvrabec@wrabco.englab.brq.redhat.com>
Date:   Mon Nov 3 17:58:30 2008 +0100

    Initial commit
```

2011 First commit to SCAP Security Guide, hardening guidance + policy references Colloquially, “SCAP Content”

```
commit 540a78f26191a69651a167d256b5af47fd3eb983
Author: Jeff Blank <blank@eclipse.ncsc.mil>
Date:   Wed Jun 8 18:45:05 2011 -0400

    added a README
```

Foreman
OpenSCAP

Ruby Gem
OpenSCAP

Puppet
OpenSCAP

SCAPtimony


HARRIS


CENTRAL INTELLIGENCE AGENCY
UNITED STATES OF AMERICA


TRESYS
TECHNOLOGY


DEPARTMENT OF JUSTICE
FEDERAL BUREAU OF INVESTIGATION


NIST
National Institute of
Standards and Technology
U.S. Department of Commerce


GOVREADY


DEFENSE INFORMATION SYSTEMS AGENCY


U.S. Army Research Laboratory


NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA


OAK
RIDGE
National Laboratory

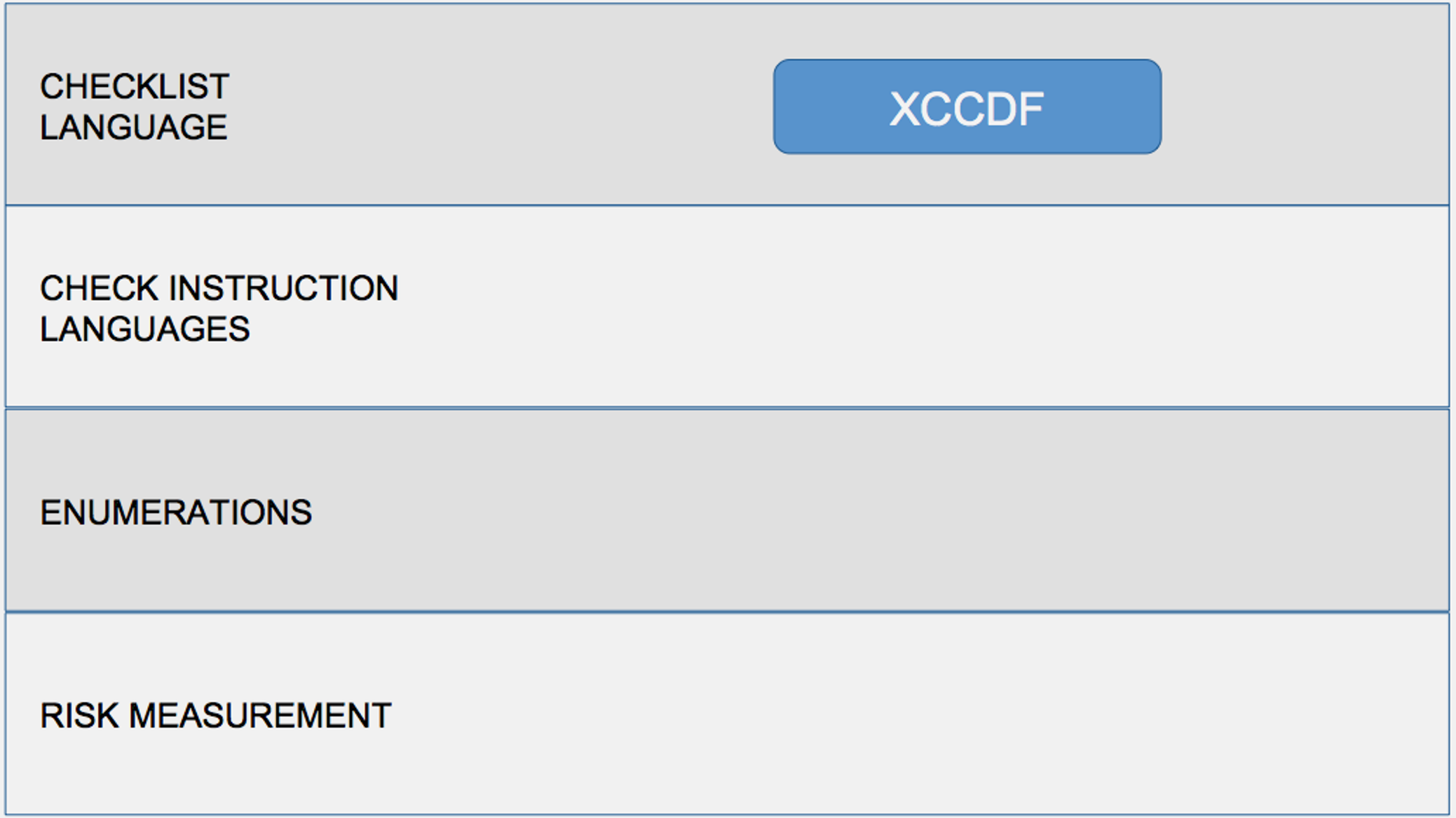

LOCKHEED MARTIN

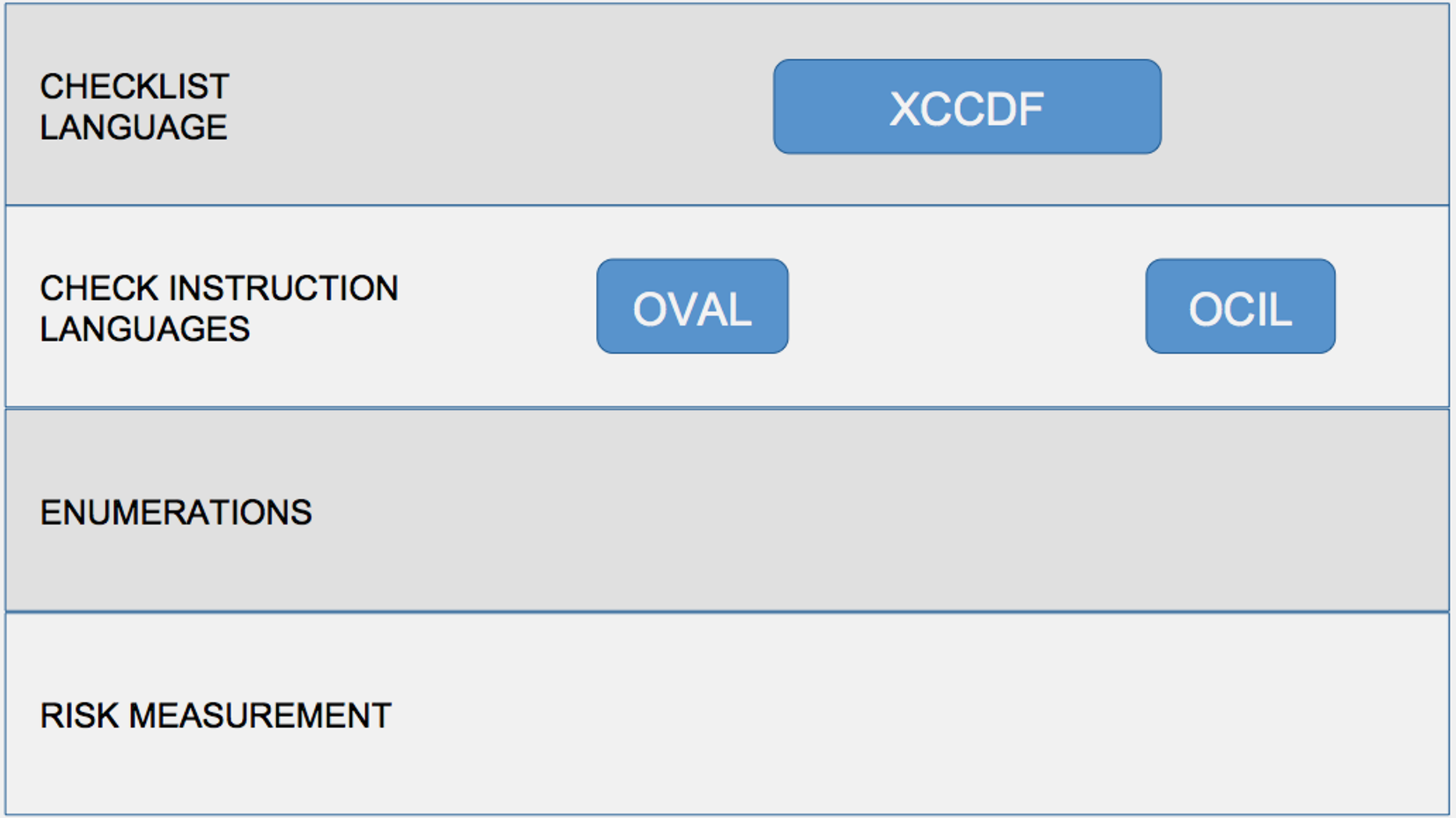
CHECKLIST
LANGUAGE

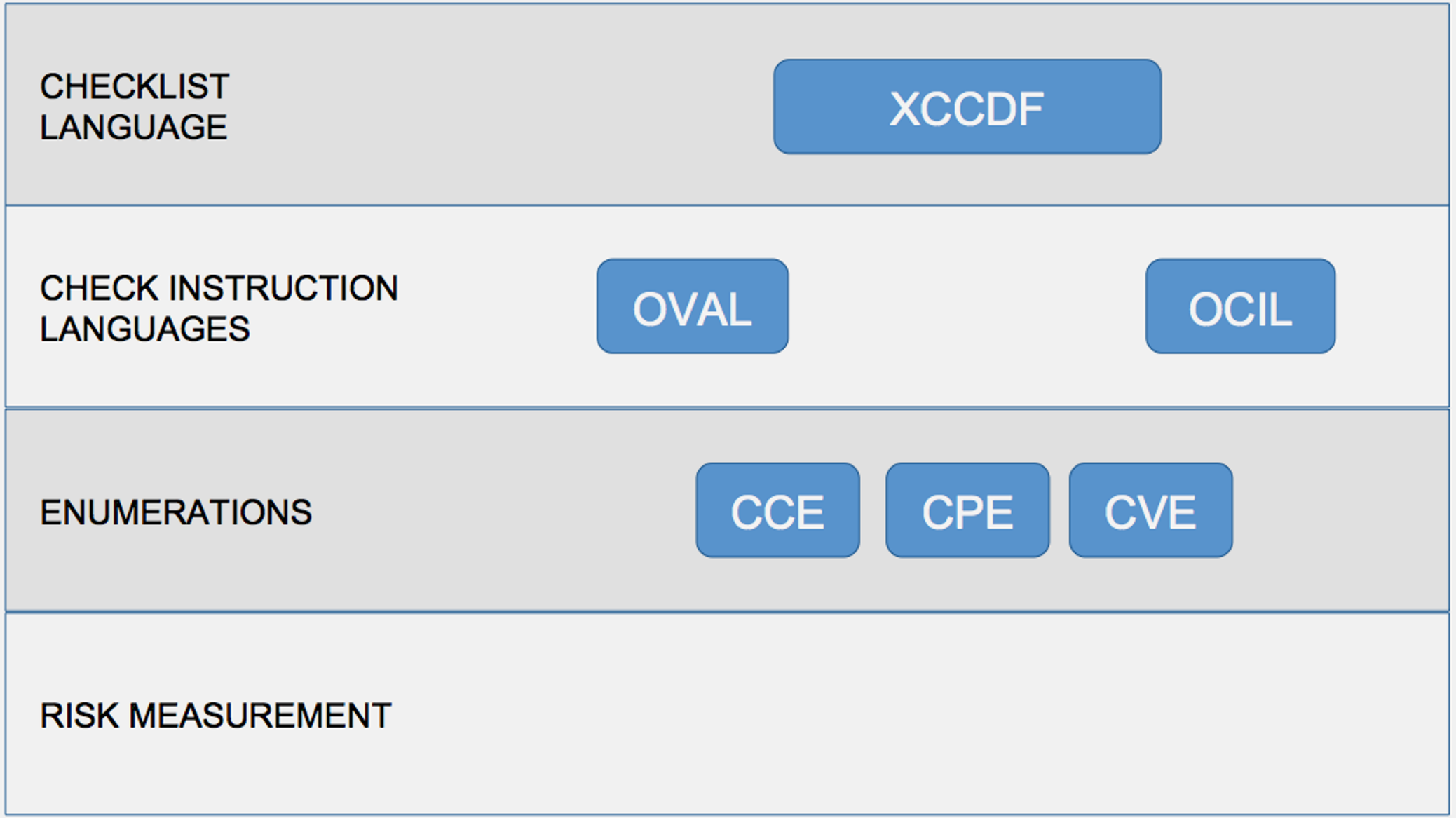
CHECK INSTRUCTION
LANGUAGES

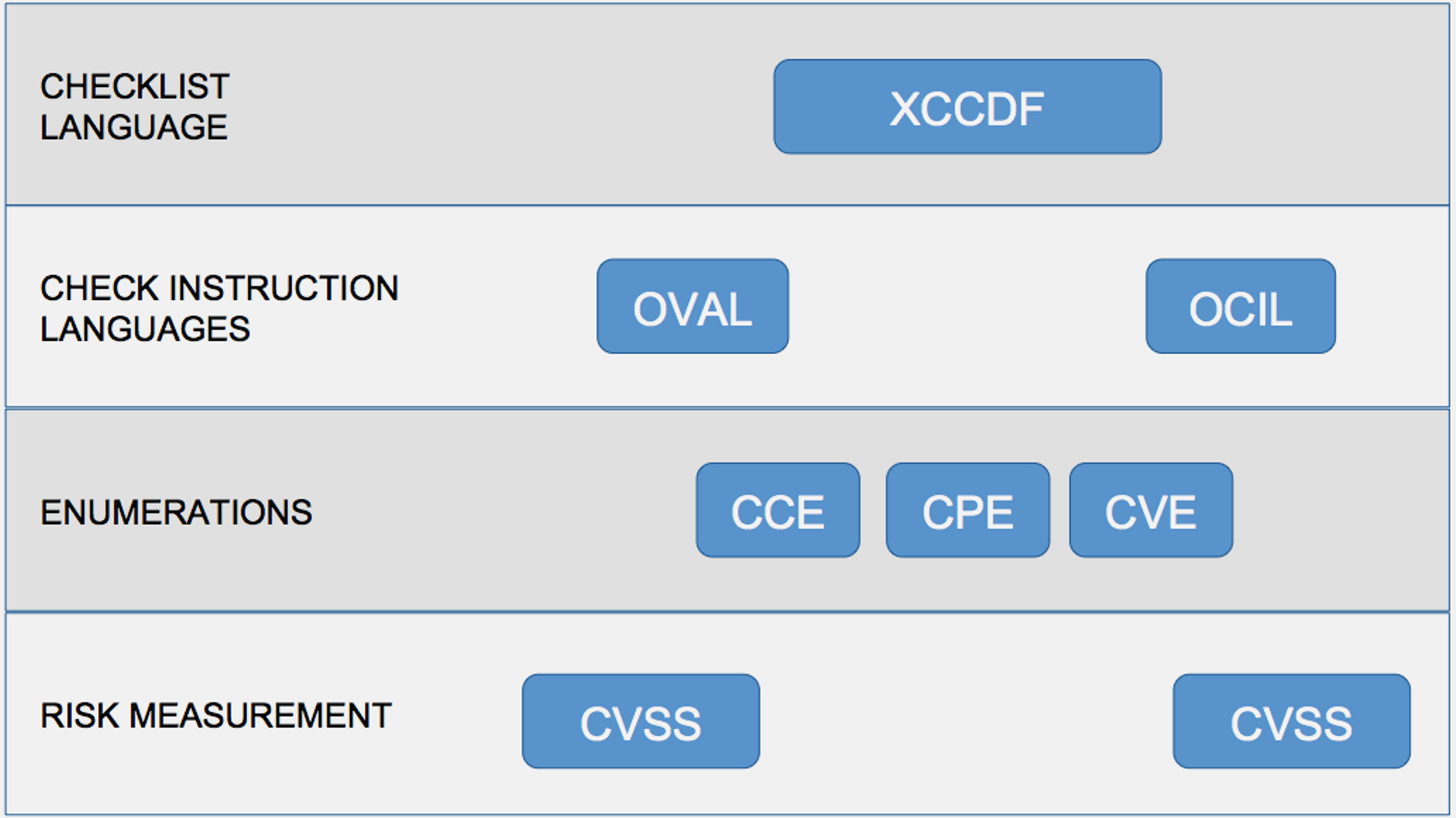
ENUMERATIONS

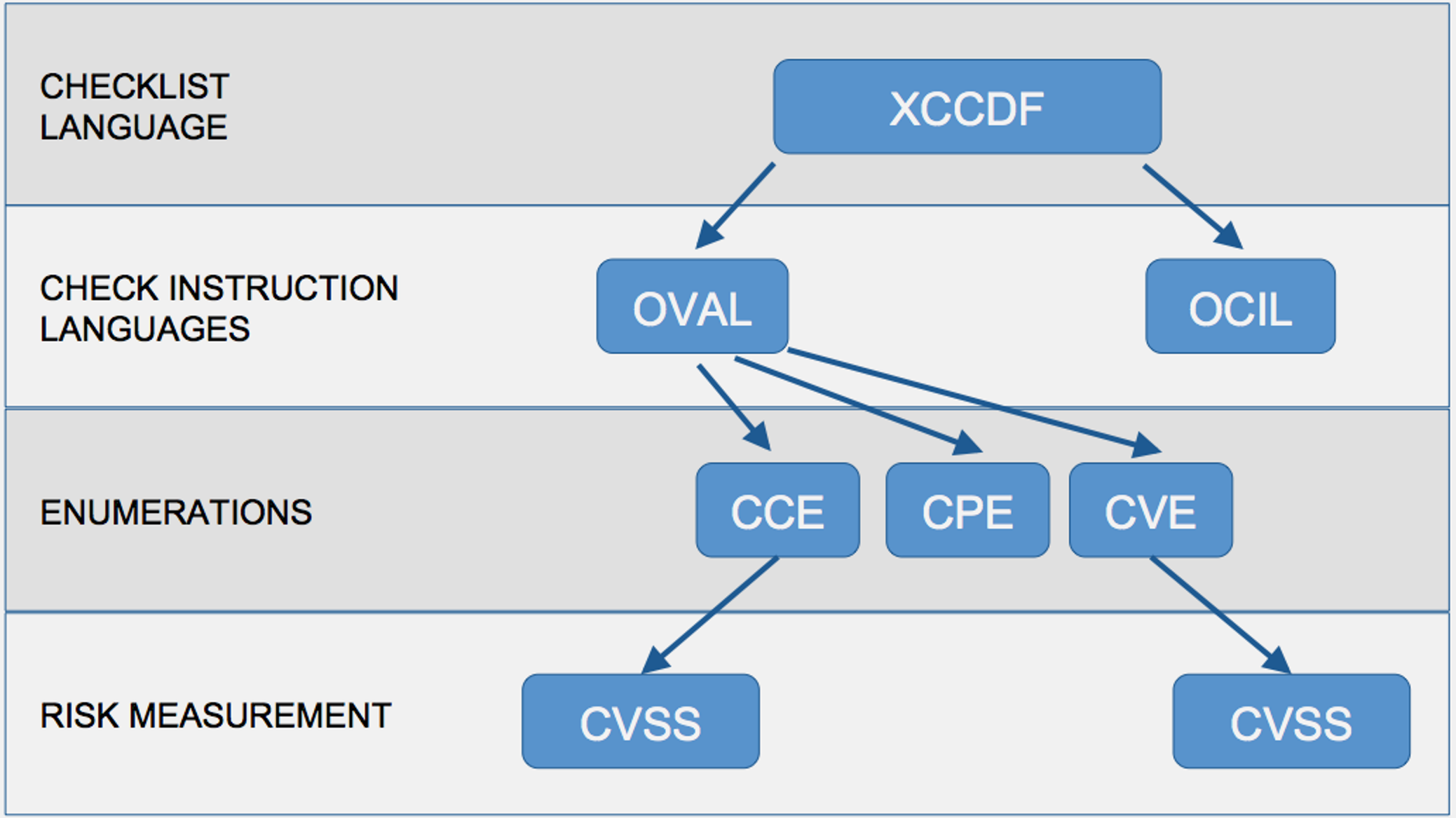
RISK MEASUREMENT


CHECKLIST
LANGUAGE
XCCDF
CHECK INSTRUCTION
LANGUAGES
ENUMERATIONS
RISK MEASUREMENT


CHECKLIST LANGUAGE
XCCDF
CHECK INSTRUCTION LANGUAGES
OVAL
OCIL
ENUMERATIONS
RISK MEASUREMENT


CHECKLIST LANGUAGE
XCCDF
CHECK INSTRUCTION LANGUAGES
OVAL
OCIL
ENUMERATIONS
CCE
CPE
CVE
RISK MEASUREMENT


CHECKLIST LANGUAGE
XCCDF
CHECK INSTRUCTION LANGUAGES
OVAL
OCIL
ENUMERATIONS
CCE
CPE
CVE
RISK MEASUREMENT
CVSS
CVSS


CHECKLIST LANGUAGE
XCCDF
CHECK INSTRUCTION LANGUAGES
OVAL
OCIL
ENUMERATIONS
CCE
CPE
CVE
RISK MEASUREMENT
CVSS
CVSS


WE DON'T ALWAYS TEST OUR CODE
BUT WHEN WE DO, IT'S LIVE
memegenerator.net

DEMO #1: INSTALL, REVIEW PROFILES

Install OpenSCAP and SCAP Content

$ sudo yum install openscap-scanner scap-security-guide

What default profiles exist?

$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel7-xccdf.xml

....

Profiles:

    pci-dss
    rht-ccp
    common
    stig-rhel7-server-upstream

....

DEMO #2: REVIEW HARDENING GUIDES

Review manpage

$ man scap-security-guide

Review HTML gudes

$ ls -l /usr/share/doc/scap-security-guide/rhel7-guide.html

DEMO #3: LOCAL SCAN, REVIEW RESULTS

Perform 1st Scan

```
$ sudo oscap xccdf eval --profile rht-ccp \
--results /root/afternoon-results.html \
--report /root/afternoon-report.xml \
/usr/share/xml/scap/ssg/content/ssg-rhel7-xccdf.xml
```

Review Results

```
$ ${web_browser} /root/afternoon-results.html
```

DEMO #4: REMEDIATION

Generate remediation scripts from results

```
$ sudo oscap xccdf generate fix \
--result-id xccdf_org.open-scap_testresult_rht-ccp \
/root/afternoon-results.xml
```

Or, remediate automatically (*be careful - no "undo"!*)

```
$ sudo oscap xccdf eval --profile rht-ccp \
--results /root/afternoon-results.xml \
--report /root/afternoon-report.xml \
--remediate \
/usr/share/xml/scap/ssg/content/ssg-rhel7-xccdf.xml
```

DEMO #5: SCAP WORKBENCH

Download SCAP Workbench

$ sudo yum -y install scap-workbench

Much of this demo is live. For extra details, https://open-scap.org