Secstate: Flexible Lockdown, Auditing, and Remediation

Certifiable Linux Integration Project
Tresys Technology
Karl MacMillan <kmacmillan@tresys.com>

Topics

- Secstate Overview
- Sample session illustrating tool usage
- Puppet / SCAP integration
- Future Plans

Secstate Overview

- Tool for security management on Linux / Unix
- Written in Python
- Automates three primary security tasks
  - Audit & Report: rapid, automated security state assessment
  - Remediate: modify (lockdown) system to put it in a compliant state
  - Maintain: maintain the system in a compliant state
- Basic operation: manages a repository of content
  - Content consists of SCAP and Puppet
  - Aligns Puppet and SCAP to automate remediation
- Primary advantages
  - Standards-based: uses NIST SCAP standards including OVAL and XCCDF
  - Model driven: users describe secure state *not* actions
  - System configuration management compatible
    - Uses Puppet internally – a widely used system management tool
  - User extensible: import new requirements and tweak existing
  - Open source and widely available

Secstate Usage

root@localhost:~

File Edit View Search Terminal Help

```
[root@localhost ~]# secstate help

/usr/bin/secstate <sub-command> [options]
Sub-commands:
    import          --  Validates and imports a benchmark and its associated OVAL into the system
    remove          --  Removes content that had been added via the 'import' command
    export          --  Exports imported content along with any changes that have been made
    select          --  Sets a portion of the benchmark (group/rule) to be selected
    deselect        --  Sets a portion of the benchmark (group/rule) to be deselected
    save            --  Saves changes made to the current benchmark to a named profile
    list            --  List the imported benchmarks
    show            --  Shows the information associated with a group, rule, or definition id
    search          --  Search through imported content
    mitigate        --  Mark a rule as mitigated and attach a remark
    remediate       --  Bring the local system i[illegible]ompliance with a benchmark
    audit           --  Audit the local system [illegible]ecified XCCDF/OVAL content
[root@localhost ~]#
```

Secstate is currently a command line tool for flexibility – graphical versions are planned. This screenshot is showing the available commands and usage.

Listing All Groups and Rules

root@localhost:~

File Edit View Search Terminal Help

```
[root@localhost ~]# secstate list -a -r
[X]Benchmark - ID: unix-stig, Title: 'Unix STIG v5r1', Profile: 'Custom'
        [X]Group - ID: unix-stig-password-controls, Title: 'Password Controls'
                [ ]Rule - ID: GEN000540-A, Title: 'Minimum days between password changes (login.defs)'
                [ ]Rule - ID: GEN000540-B, Title: 'Minimum days between password changes (shadow)'
                [ ]Rule - ID: GEN000560, Title: 'Password is assigned'
                [ ]Rule - ID: GEN000580-A, Title: 'Password Minimum Length (login.defs)'
                [X]Rule - ID: GEN000580-B, Title: 'Password Minimum Length (PAM)'
                [ ]Rule - ID: GEN000600-A, Title: 'Password Minimum Alphabetic Characters'
                [ ]Rule - ID: GEN000600-B, Title: 'Password Minimum Uppercase Characters'
                [ ]Rule - ID: GEN000620, Title: 'Password Minimum Numerics'
                [ ]Rule - ID: GEN000640, Title: 'Password Minimum Special Characters'
                [ ]Rule - ID: GEN000680, Title: 'Passwords Cannot Repeat Characters'
                [ ]Rule - ID: GEN000700-A, Title: 'Maximum Days Between Password Changes (login.defs)'
                [ ]Rule - ID: GEN000700-B, Title: 'Max      Days Between Password Changes (shadow)'
                [ ]Rule - ID: GEN000720, Title: 'Max        tween Root Password Changes (shadow)'
                [ ]Rule - ID: GEN000740
                [ ]Rule - ID: GEN000760
                [ ]Rule - ID: GEN000780
                [ ]Rule - ID: GEN000800
                [ ]Rule - ID: GEN000820
                [ ]Rule - ID: GEN000840
                [ ]Rule - ID: GEN000860
[root@localhost ~]#
```

Adding the “-a” command causes all groups and rules to be displayed. Here is a partial list of the Unix STIG to demonstrate (this example is abbreviated to make the display more manageable).

Auditing System State

root@localhost:~

File Edit View Search Terminal Help

```
[root@localhost ~]# secstate audit
--Results for 'unix-stig' (Profile: 'Custom')--
Passed:         0
Mitigated:      0
Failed:         1
Fixed:          0
Not Selected:   19
Not Checked:    0
Not Applicable: 0
Error:          0
Informational:  0
Unknown:        0

[root@localhost ~]#
```

The audit command inspects the current system state according to the active benchmarks and rules. It outputs a summary and saves the results in a standardized XML format and easy to read HTML format.

Here the minimum password length configuration check is failing.

HTML Audit Output

SecState Audit Results - Mozilla Firefox

Failures: 1

⊟ FAILURE 1: GEN000580-B - A password minimum length must be specified in the PAM configuration.

FixText : Add the option minlen=# to the pam_cracklib.so module entry in /etc/pam.d/system-auth.

(PAM) Password Complexity - Minimum Length (oval:com.tresys.oval.rhel:def:1014)

1. (PAM) Verify the password minimum length meets or exceeds the specified length [**false**] (oval:com.tresys.oval.rhel:tst:1015)

Operating System (OS):	Linux	Failures:	1
OS Version:	#1 SMP Tue Aug 17 22:54:19 UTC 2010	Mitigations:	0
Architecture:	i686	Passes:	0
		⊞ Other	19

⊞ **Interfaces**

Failures: 1

⊟ FAILURE 1: GEN000580-B - A password minimum length must be specified in the PAM configuration.

FixText : Add the option minlen=# to the pam_cracklib.so module entry in /etc/pam.d/system-auth.

(PAM) Password Complexity - Minimum Length (oval:com.tresys.oval.rhel:def:1014)

1. (PAM) Verify the password minimum length meets or exceeds the specified length [**false**] (oval:com.tresys.oval.rhel:tst:1015)

check: **all**

check_existence: **all_exist**

State

⊟ value

⊟ external_variable

pass-min-length-var: **Password Minimum Length**

value: **8**

Tested Items

⊟ variable_item [**false**]

value: **7**

var_ref: **oval:com.tresys.oval.r**

Mitigations: 0

Passes: 0

Done

This is the HTML output showing the same failure and some additional system information.

Remediation

root@localhost:~

File Edit View Search Terminal Help

```
[root@localhost ~]# secstate remediate -x audit-results/
-- Remediating unix-stig --
notice: //passreqs/Pam[password:pam_cracklib.so]/module_args: module_args changed 'minlen=7 retry=3 try_first_pas
s type=' to 'minlen=8 retry=3 try_first_pass type='
[root@localhost ~]#
```

The remediate command takes a set of failures (from a results XML file) and changes the system configuration so that compliance is achieved.

In addition to automated remediation (using Puppet), the XCCDF content can list manual remediation steps.

Verification of Remediation

root@localhost:~

File Edit View Search Terminal Help

```
[root@localhost ~]# secstate audit
--Results for 'unix-stig' (Profile: 'Custom')--
Passed:         1
Mitigated:      0
Failed:         0
Fixed:          0
Not Selected:   19
Not Checked:    0
Not Applicable: 0
Error:          0
Informational:  0
Unknown:        0

[root@localhost ~]#
```

After verification, audit can be run again to verify success. Here the password GEN is shown passing.

Core Use Cases and Features

- Remediation
  - Manual, administrator driven
  - Automated based upon scans
  - Full configuration management (Puppet master)
- Customization of security requirements
  - Importing security benchmarks
  - Disabling individual rules
  - Setting key variables
- All with integration of SCAP and Puppet

System Configuration Management

- Security and management tools often conflict
  - Both sets of tools change configuration
  - Lack of integration results in conflicts
  - System state described in multiple places
- System configuration management increasing
  - Data centers are increasingly automated
  - Higher quality with fewer administrators
  - Virtualization / cloud driving adoption
  - Need for integration with security lockdown is increasing
- Secstate aims to unify management and lockdown
  - Security and general configuration treated identically
  - Uses mature system management tool internally (Puppet)
  - Can integrate with enterprise Puppet systems
  - Other configuration management tools can be integrated

Notes on SCAP

- SCAP has many advantages
  - Viable cross-platform security auditing
  - Increased automation for *many* tasks
- Unfortunately SCAP is not perfect
  - Complex, layered set of standards
    - CCE, CPE, CVE, OVAL, XCCDF, . . .
    - Difficult to push customization through all the layers
  - Languages tend to be challenging
    - Seems to emphasize *machine* readable
    - Verbose, obfuscated syntax
  - OVAL probes are very limiting
    - Lack of language features for abstractions
    - Too much becomes textfilecontent54 (especially on Linux)
    - But probes offer safety and predictability
  - Central management of content
    - Need for private namespaces
    - Large body of high-quality content has yet to emerge

XCCDF Example – Password Length

```
<Rule id="pass-min-length" selected="1">
        <title>GEN0000580 - Password Minimum Length</title>
                <description> A password minimum length must be
                                        specified.</description>
        <fix system="urn:xccdf:fix:script:puppet">
          class : passreqs
          parameter : login_defs_min_len : <sub idref="pass-min-length-var" />
        </fix>
        <check
         system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
                <check-export value-id="pass-min-length-var“
                  export-name="oval:com.tresys.oval.rhel:var:1017"/>
                <check-content-ref href="passreqs.oval.xml“
                  name="oval:com.tresys.oval.rhel:def:1014"/>
        </check>
</Rule>
```

XCCDF Values

```
<Value id="pass-min-length-var" type="number“
  operator="greater than or equal">
        <title>Password Minimum Length</title>
        <description>
           Contains the specified minimum length of passwords for the
           system.
        </description>
        <value>8</value>
</Value>
```

OVAL Example

```
<definition class="compliance" id="oval:com.tresys.oval.rhel:def:1014"
version="1">
        <metadata>
            <title>(PAM) Password Complexity - Minimum Length</title>
            <affected family="unix">
                <platform>Red Hat Enterprise Linux 5</platform>
            </affected>
            <reference ref_id="GEN000580" source="UNIX STIG" />
            <description>Password Complexity</description>
        </metadata>
        <criteria>
            <criterion test_ref="oval:com.tresys.oval.rhel:tst:1015" />
        </criteria>
</definition>
```

Eventually . . . Object

```
<textfilecontent54_object id="oval:com.tresys.oval.rhel:obj:1022"
version="1" xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-
5#independent">
        <path>/etc</path>
        <filename>login.defs</filename>
        <pattern operation="pattern match">
                ^[^#]*PASS_MIN_LEN[[:space:]]+([[:digit:]]+)
        </pattern>
        <instance datatype="int" operation="greater than or equal">
                1
        </instance>
</textfilecontent54_object>
```

Addressing OVAL Language Woes

- Developed SCC to generate OVAL
  - New language with simpler syntax
  - Maps directly to OVAL semantics
- Tools approach for simplifications
  - Focus on UI – seldom address real issues
  - Often force a particular workflow
- Language approach flexibly addresses challenges
  - Focuses on core issues without forcing a particular workflow
  - Surprisingly easier to maintain compiler than tools
  - Appropriate for likely OVAL authors
- Key OVAL challenges solved by SCC
  - Verbosity – SCC is compact and expressive
  - IDs – SCC provides *human* readable IDS w/ stable mappings
  - Locality – related statements grouped together
  - Mapping – simple, predictable mapping to OVAL

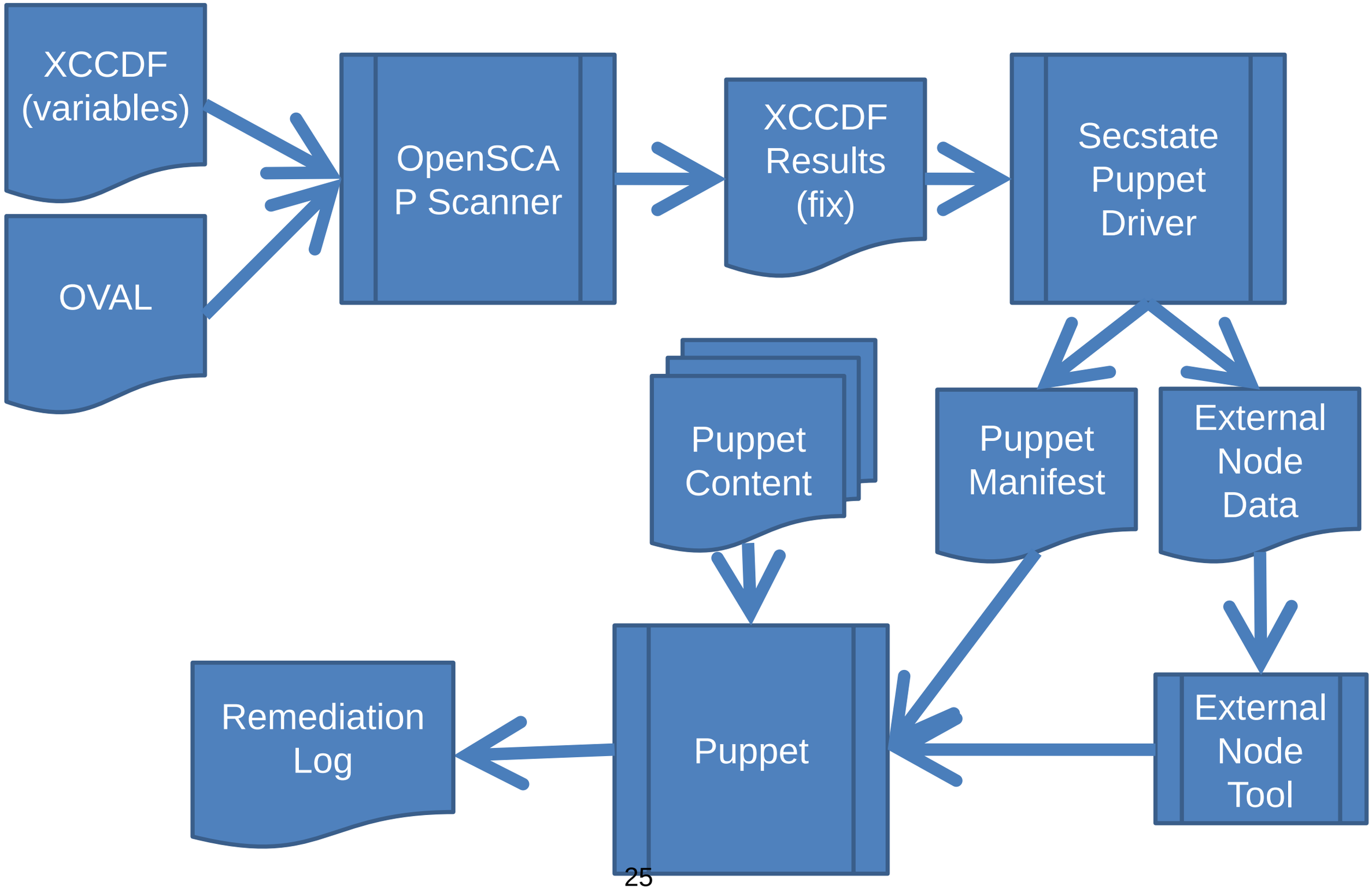
SCC Example

```
test ind:variable pam-pass-min-len {
    @check="all"
    @comment="(PAM) Verify the password minimum length meets or exceeds the specified length"
    object { variable<=pam-pass-minlen-var }
    state { value { @datatype="int" @operation="greater than or equal" variable<=extern-pass-minlen-var } }
}
object ind:textfilecontent54 cracklib-pass-minlen {
                    @comment="Cracklib library for PAM"
                    path="/etc/pam.d"
                    filename="system-auth"
                    pattern="^[^#]*password.*(?:required|requisite).*pam_cracklib\.so.*minlen=-?(\d+).*" {
                                @operation="pattern match"}
                    instance="1" { @operation="greater than or equal" @datatype="int" }
}
variable int:external extern-pass-minlen-var {
    @comment="Obtains the minimum length specified externally"
}
variable int:local pam-pass-minlen-var {
    @comment="Contains the pam password minlen"
    object_component { object<=cracklib-pass-minlen @item_field="subexpression" }
}
```

Puppet / SCAP Integration Challenges

- Remediation only performs partial configuration
  - Only failed configuration is performed
  - Requires aligning scan rules and Puppet
- Puppet and the unknown
  - Puppet designed to fully specify state
    - e.g., set complete file mode on a list of files
    - Easier to work with templated configuration files
  - Security requirements often broad
    - All filesystems mounted nosuid
    - Ensure man pages have perms set to 644
  - Requires custom Puppet providers
- Customization in a single place
  - Desire to custom requirements once (e.g., min passwd length)
  - Have that impact both Puppet and SCAP

Basic Process (Single System)



Key Integration Points

- XCCDF Fix tag
  - Specifies Puppet classes and variables
  - Each rule contains a fix element
  - Fine-grained mapping of XCCDF to Puppet
- External nodes tool
  - Synchronization mechanism for customization
  - Transfers XCCDF variables to Puppet
- Puppet driver
  - Instantiates needed Puppet classes
  - Runs Puppet commandline tool
- Requires tailored SCAP *and* Puppet
  - For best results – other content still usable
  - Content still standard – no language extensions required

XCCDF Example – Password Length

```
<Rule id="pass-min-length" selected="1">
        <title>GEN0000580 - Password Minimum Length</title>
                <description>
                        A password minimum length must be specified.
                </description>
        <fix system="urn:xccdf:fix:script:puppet">
          class : passreqs
          parameter : login_defs_min_len : <sub idref="pass-min-length-var" />
        </fix>
        <check
          system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
                <check-export value-id="pass-min-length-var“
                  export-name="oval:com.tresys.oval.rhel:var:1017"/>
                <check-content-ref href="passreqs.oval.xml“
                  name="oval:com.tresys.oval.rhel:def:1014"/>
        </check>
</Rule>
```

Puppet Example

```
 if $shadow_max_days != '' {
      exec { "for shadowname in `awk -F: '{ print \$1 }' /etc/shadow`;
do passwd -x $shadow_max_days \$shadowname; done" :
          path => "/bin:/usr/bin"
      }
 }
 if $login_defs_min_len != '' {
      exec { "sed -i -e '/PASS_MIN_LEN/d' -e '$
a\\PASS_MIN_LEN=$login_defs_min_len' /etc/login.defs" :
          onlyif => "test -f /etc/login.defs",
          path => "/bin:/usr/bin"
      }
   }
```

Future Plans

- Port to additional systems
  - Current target is Fedora
  - Port to RHEL 5 is needed (and straightforward)
  - FY11 official support for RHEL4,5,6 and port to Solaris 10 with TX
  - FY11 remote reporting
  - FY12 port to STOP 7 and Solaris 11 with TX
  - FY12 remote policy update and execution
- Additional requirement sets
  - Current target is the Unix STIGS
  - Desired requirements: other STIGS, CNSS 1253, NIST SP 800-53 rev3, DCID 6/3
- Usability and documentation
  - User and developer documentation expansion
  - Graphical configuration tools (FY11)

Questions?

https://fedorahosted.org/secstate/

http://www.tresys.com

BACKUP