TRESYS TECHNOLOGY

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
- SCAP Introduction
- Puppet / SCAP integration
- Future Plans

Secstate Overview

- Tool for security management on Linux / Unix
- Automates three primary security tasks
  - Lockdown: create secure systems install to end-of-life
  - Audit: rapid, automated security state assessment
  - Remediation: correct configuration errors
- Primary advantages
  - Standards-based: uses NIST SCAP
  - 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@j4cfy41:~

File Edit View Terminal Tabs Help

```
[root@j4cfy41 ~]# secstate --help
usage: secstate <command> [flags]

commands:
    import          Validates and imports a benchmark and its associated oval into the system
    select          Sets a portion of the benchmark (group/rule) to be selected
    deselect        Sets a portion of the benchmark (group/rule) to be deselected
    list            List the imported benchmarks
    show            Shows the information associated with a group, rule, or definition id
    search          Search through imported content
    remediate       Remediate issues.
    audit           Audit the local system based the specified benchmark
[root@j4cfy41 ~]#
```

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@j4cfy41:~
File  Edit  View  Terminal  Tabs  Help
[root@j4cfy41 ~]# secstate list -r -a
[X]Benchmark - Title: "Unix STIG v5r1", ID: tresys-unix-stig
    [X]Group - Title: "Password Controls", ID: tresys-password-controls
        [ ]Rule - Title: "Change no more than", ID: gen000540
        [ ]Rule - Title: "Password is assigned", ID: gen000560
        [X]Rule - Title: "Minimum length", ID: gen000580
        [ ]Rule - Title: "Minimum alphabetic characters", ID: gen000600
        [ ]Rule - Title: "Minimum numeric characters", ID: gen000620
        [ ]Rule - Title: "Minimum special characters", ID: gen000640
        [ ]Rule - Title: "No guessable information", ID: gen000660
        [ ]Rule - Title: "No consecutive characters", ID: gen000680
        [ ]Rule - Title: "Change at least", ID: gen000700
        [ ]Rule - Title: "Change root at least", ID: gen000720
        [ ]Rule - Title: "Change non-interactive at least", ID: gen000740
        [ ]Rule - Title: "Lock if inactive for"  gen000760
        [ ]Rule - Title: "Not easily guessabl  000780
        [ ]Rule - Title: "Cannot
        [ ]Rule - Title: "Global p
        [ ]Rule - Title: "Limited
        [ ]Rule - Title: "Change r
[root@j4cfy41 ~]#
```

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@j4cfy41:~
File  Edit  View  Terminal  Tabs  Help
[root@j4cfy41 ~]# secstate audit --xml results.xml --html results.html
Title: "Minimum length", ID: gen000580, Result: false
[root@j4cfy41 ~]#
```

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

OVAL Results - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///root/results.html

Most Visited Red Hat Red Hat Magazine Red Hat Network Red Hat Support

TRESYS TECHNOLOGY

XCCDF Benchmark Results

True | False | Error | Unknown | Not Applicable | Not Evaluated

ID	Result	Class	Reference ID	Title
gen000580	false	compliance	GEN000580	Minimum Length

System Information

Host Name	j4cfy41
Operating System	Linux
Operating System Version	#1 SMP Tue Aug 18 15:51:48 EDT 2009
Architecture	x86_64
Interface Name	lo
IP Address	127.0.0.1
MAC Address	00:00:00:00:0
Interface Name	eth0
IP Address	fe80::20c:29ff:
MAC Address	00:0C:29:44:D

Done

XCCDF Benchmark Results

True | False | Error | Unknown | Not Applicable | Not Evaluated

ID	Result	Class	Reference ID	Title
gen000580	false	compliance	GEN000580	Minimum Length

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

Remediation

```
root@j4cfy41:~
File  Edit  View  Terminal  Tabs  Help
[root@j4cfy41 ~]# secstate remediate -i results.xml
Title: "Minimum length", ID: gen000580, Result: remediated
[root@j4cfy41 ~]#
```

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@j4cfy41 ~]# secstate audit --xml final-results.xml --html final-results.html
Title: "Minimum length", ID: gen000580, Result: true
[root@j4cfy41 ~]#
```

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

SCAP Introduction

- NIST SCAP is a standard for security description
  - Family of XML-based languages
  - Covers a large variety of security information
    - Requirements (XCCDF), Auditing (OVAL), Vulnerabilities (CVE)
- Standardizes security description *and* reporting
  - Reports can be machine processed, summarized, and searched
  - Potential to ease C&A artifact creation and updating
- Mandated for use on many government systems
  - Required on all federal desktops (part of FDCC)
  - All HBSS systems consume SCAP
- Growing adoption outside of government
- Enables vendor neutral security scanning
  - SCAP validated tools are interoperable
  - Eliminates vendor lockdown for security auditing

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, . . .
  - Languages tend to be challenging
    - Seems to emphasize *machine* readable
    - Verbose, obfuscated syntax

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:]]+)</patter
    n>
        <instance datatype="int" operation="greater than or
    equal">1</instance>
 </textfilecontent54_object>
```

Addressing SCAP 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
- 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

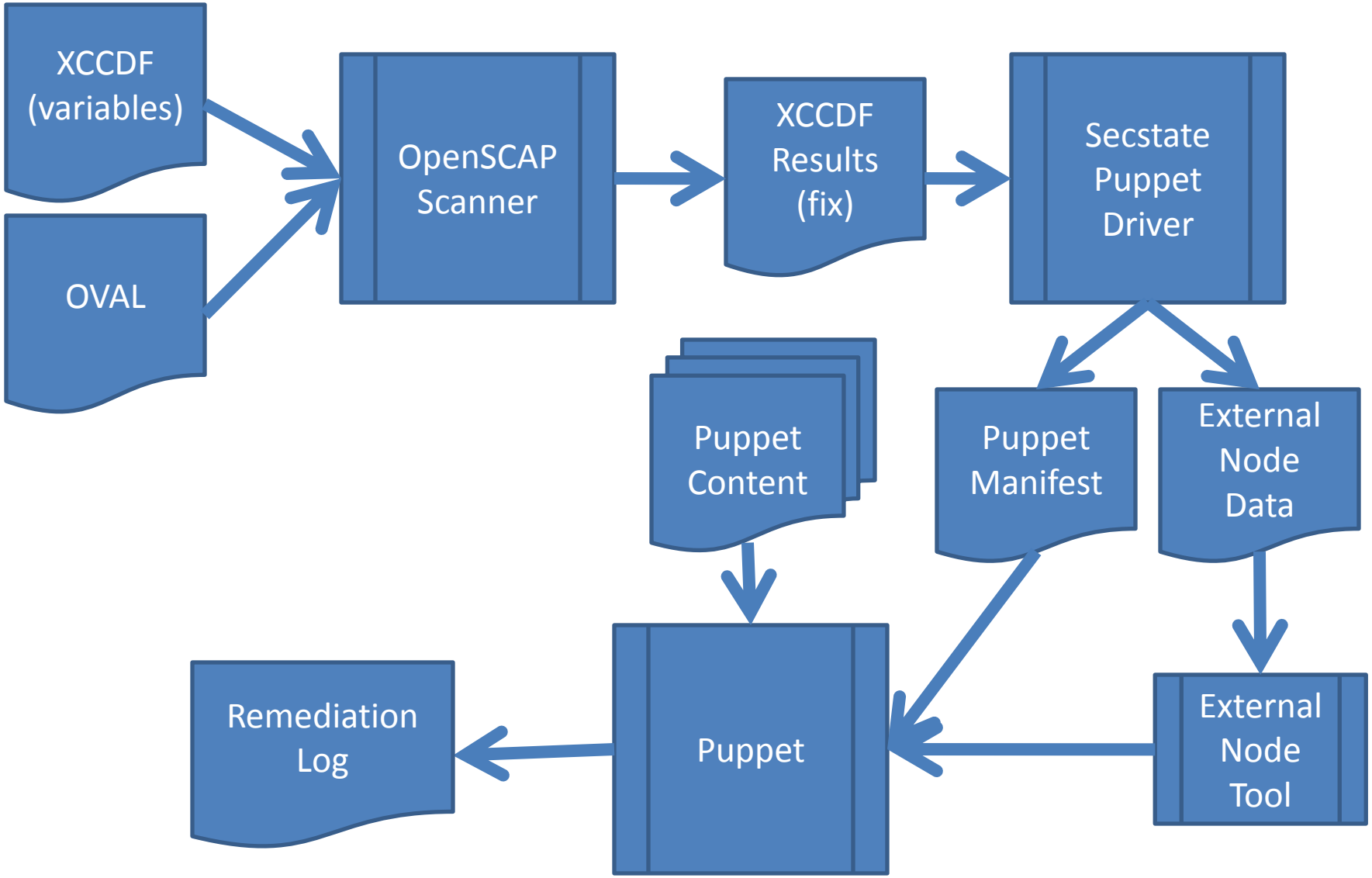
SCC Example

```
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
  - 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)
  - Other systems possible – Solaris, STOP, etc.
- Additional requirement sets
  - Current target is the Unix STIGS
  - Desired requirements: other STIGS, 1253, 800-53
- Usability and documentation
  - User and developer documentation expansion
  - Graphical configuration tools

TRESYS TECHNOLOGY

Questions?

https://fedorahosted.org/secstate/

http://www.tresys.com