Tools for auditing and securing cloud systems

Golden Images

Tools like OpenSCAP and Aqueduct can make life much easier when implementing cloud security standards. *By Kurt Seifried*

One thing I love about cloud computing is being able to deploy a dozen or more systems with the click of a button. Testing and deploying larger scale systems has never been so easy. Additionally, I don't have to deal with hardware failures. Rather than driving out to the location at 2am to fix a dead server, I simply launch a new instance, or stop and start an instance, which on many cloud providers results in the use of a new physical server. Two clicks and a two-minute wait versus driving downtown at 2am is not a hard choice to make (especially in a Canadian winter). Also, I can turn off and stop paying for unneeded servers, which my accountant approves of.

A lot of businesses, governments, and even the military seem to be following suit. Amazon has deployed GovCloud in the United States, which is tailored for US government use and doesn't allow non-government users on the service. How do you audit and secure all these systems? You start with a golden image – an installation of the OS that is configured and locked down in advance.

As the next step, you can create updated golden images every time a security update comes out and restart all the servers. In addition to being able to install security patches and updates, you will also need to detect and change configurations as needed. For example, the Apache Range Header Denial of Service had a simple workaround that prevented exploitation, but manually logging in, modifying `httpd.conf`, and restarting Apache on a few hundred servers is not my idea of a good time.

Security Standards

Another aspect of computer security is the difficulty in coming up with standards that address all the potential problems. Do you need a login banner stating that unauthorized access is prohibited and that activity must conform to a terms-of-service or acceptable-use policy? I don't know, it depends on your jurisdiction. What about all the services running by default on a modern system? Is Avahi safe to leave running? What about rpcidmapd? What about dm-event, sm-client, or mcelog?

The good news is that governments around the world have put real thought into these problems and come up with security standards. Examples include the Security Technical Implementation Guide (STIG) [1], which is put out by the Defense Information Systems Agency and covers everything from Active Directory to Firefox and Apache HTTPD. Another good one is the US Government Configuration Baseline (USGCB). Although it tends to focus on Windows, it has some excellent content for Linux (Red Hat 5 in particular) [2], including a Kickstart script that will give you a largely secure-by-default installation. The other advantage of many US government standards is that they not only contain the general security goal (e.g., "enforce strong passwords"), but they also contain details of exactly what you need to do on various operating systems to accomplish this.

SCAP

The Security Content Automation Protocol (SCAP) [3] is the big player in town now for government security. SCAP is actually a com-

KURT SEIFRIED

Kurt Seifried is an Information Security Consultant specializing in Linux and networks since 1996. He often wonders how it is that technology works on a large scale but often fails on a small scale.

bination of various security protocols and efforts that include CCE (which handles configuration of systems), CWE (which handles identification of vulnerability types, and XCCDF (which can be used to create security checklists for people and software use).

The SCAP standard is also designed to prevent vendor lock in, ending the lock-in of proprietary security information formats (e.g., custom XML definitions or weird binary formats) and software. It is also a growing effort – ultimately, SCAP should provide a complete set of tools, data formats, information formats, security standards, and so forth that can be used to secure systems and automate auditing, compliance, and remediation.

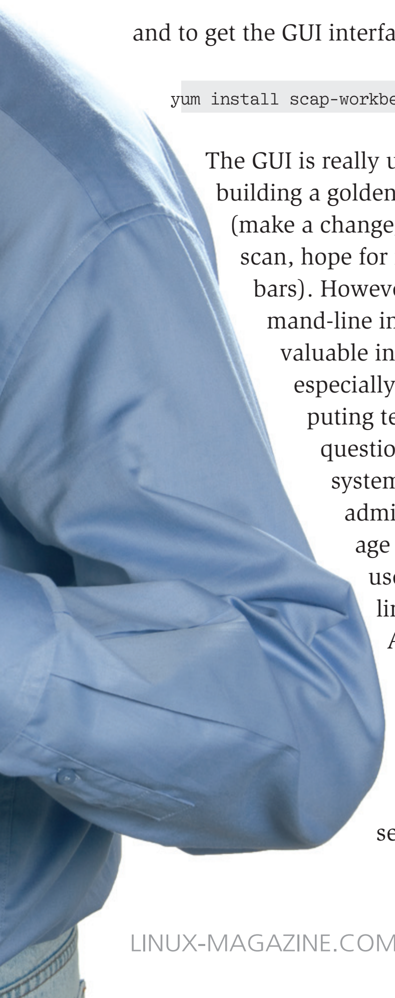
OpenSCAP

An open source implementation of SCAP is also available. OpenSCAP [4] provides a set of libraries to handle SCAP document parsing, and more importantly, it provides a security scanner that can take SCAP content, scan a system for issues, and produce a nice report that can then be used to fix the problems. Installation is easy [5]; on Red Hat, simply do

```
yum install openscap openscap-content ↩
  openscap-utils
```

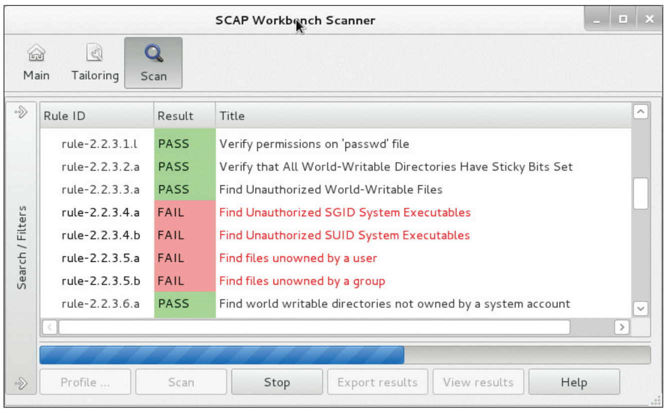
and to get the GUI interface, do

```
yum install scap-workbench
```



The GUI is really useful if you are building a golden master image (make a change, re-run the scan, hope for more green bars). However, the command-line interface is more valuable in my opinion, especially in cloud computing terms. Trick question: How many systems can a single administrator manage if they have to use a command-line interface? Answer: All of them – at the same time.

About the only downside of SCAP-based security scans is that some of the tests include things like "Find Unauthorized SGID System Executables," which basically does a full filesystem listing. This wouldn't be so bad except a number of tests trigger this behavior, so a scan can take several minutes to more than an hour on a large file server.



Rule ID	Result	Title
rule-2.2.3.1.l	PASS	Verify permissions on 'passwd' file
rule-2.2.3.2.a	PASS	Verify that All World-Writable Directories Have Sticky Bits Set
rule-2.2.3.3.a	PASS	Find Unauthorized World-Writable Files
rule-2.2.3.4.a	FAIL	Find Unauthorized SGID System Executables
rule-2.2.3.4.b	FAIL	Find Unauthorized SUID System Executables
rule-2.2.3.5.a	FAIL	Find files unowned by a user
rule-2.2.3.5.b	FAIL	Find files unowned by a group
rule-2.2.3.6.a	PASS	Find world writable directories not owned by a system account

Figure 1: The OpenSCAP workbench during a scan.

Remediating Security Problems

Once you've identified a security problem, the next step is to fix it? For some issues, this process will be easy, but for some, it won't. A great example is xinetd. Suppose, for the sake of argument, your security policy says xinetd must be disabled. How do you go about disabling it? You could use `chkconfig` to disable xinetd, but this makes it really easy to enable again at a later time.

A more permanent option would be to remove the xinetd package entirely – not much could go wrong with that, right? One problem involves the dependencies. What if an existing package depends on xinetd? Or, what if a package installed in the future requires xinetd, causing it to be installed again? Another option would be to disable each xinetd service with `disable = yes`, but this will fail quite quickly if a new service using xinetd is installed and it defaults to being enabled.

A final possibility would be to remove the xinetd binary file. Then, if any program installs an xinetd configuration with `disable = no`, you're safe, and enabling xinetd with `chkconfig` won't affect things. However, the next time you update xinetd, the binary file will be reinstalled, putting you back at square one. I'm not sure what the moral of this story is, but, in any case, I strongly recommend re-running the OpenSCAP scan any time software is installed or updated.

Aqueduct

The topic of remediation brings me to Aqueduct [6]. Aqueduct is designed to take the security information for a configuration and bring the system in line with it. This means that, for example, an SCAP report can be fed into Aqueduct, and Aqueduct will bring the system into compliance. Why not simply have OpenSCAP fix the problems? Well, for the simple reason that SCAP is not the only security protocol out there; others, such as PCI-DSS, NISPOM, CIS, and STIG, also need to be considered.

Conclusion

If you need to deal with US government standards, tools such as SCAP and STIG will be critical for you. And, if you don't need to deal with them now, it's likely the companies you work with will adopt these standards at some point (either because they work with the US government or because coming up with their own standards is simply too expensive). Additionally, if you want to implement your own security standards, using tools like OpenSCAP and Aqueduct will be much easier than building your own. ■■■

INFO

[1] Security Technical Implementation Guide: *http://iase.disa.mil/stigs/a-z.html*

[2] United States Government Configuration Baseline: *http://usgcb.nist.gov/usgcb/rhel/download_rhel5.html*

[3] Security Content Automation Protocol: *http://scap.nist.gov/*

[4] OpenSCAP: *http://www.open-scap.org*

[5] SCAP Workbench: *https://fedorahosted.org/scap-workbench/*

[6] Aqueduct: *https://fedorahosted.org/aqueduct/*