Source: *Stable Diffusion AI*

LABORATORY 6. AUTOMATABLE LINUX OS SECURITY POLICIES

DEPARTMENT OF COMPUTER SCIENCE. UNIVERSITY OF OVIEDO

Computer Security | 2022 – 2023 (V2.7 *"S-81 Isaac Peral"*)

CONTENT

THE INFRASTRUCTURE OF THIS LABORATORY

This laboratory **cannot be run in containers**, as the tools we are going to use are not designed to be run inside them. In fact, there is a special version of `oscap` designed to be run inside *Docker* containers (`oscap-docker`), but using it is out of the scope of the laboratory. Therefore, **we strongly encourage you to create a snapshot of your current virtual machine** to run the programs of this laboratory. **See the extra lab documentation for details about how to create snapshots**.

Lab 6 infrastructure folder just contains some files needed to perform the activities of this laboratory, to save you time.

BLOCK 1: MANAGING OSCAP POLICIES

Updating *OpenSCAP* security policies

Practical application: *You need to update your hardening rules of the scap-security-guide package to the latest version*

OpenSCAP Compliance as Code tool and *SCAP Workbench* (a free *OpenSCAP* GUI) **are already installed in the course virtual machine**. If you created your own machine, to install the *OpenSCAP* tool you have to follow the official installation procedures for *Ubuntu*: https://www.open-scap.org/tools/openscap-base/#download. Install also *SCAP Workbench*, with: `sudo apt install scap-workbench`

To ensure the best result with *OpenSCAP*, it is required to update its security policies. These contain validation and remediation rules for different OS and software.

- Download the latest version of these policies from the official *GitHub* repository (https://github.com/ComplianceAsCode/content). Update the version number of this link with the latest: `wget https://github.com/ComplianceAsCode/content/releases/download/v0.1.60/scap-security-guide-0.1.60-oval-5.10.zip`, or go to the *Releases* section of the *GitHub* repo *(**NOTE**: In the **Lab 6 infrastructure** files you can find a folder with a policy from this repository, if you want to use this one and avoid downloading a newer one).*
- Unzip the downloaded file and copy their contents to a folder you create. **This folder will be the one you will use later to load the latest *OpenSCAP* contents.**

Once this is done, we will have the latest available ***security profiles***. Each security profile is tailored to a specific use (organization, public entity…) and includes a different list of security controls. By selecting one of them you can check how many of its security rules your system complies with, and even automatically fix some of them. To know the available profiles for *Ubuntu 18.04* do: `sudo oscap info <unzip folder>/ssg-ubuntu1804-ds-1.2.xml`

Expected results: This activity will be finished when you download and install the most up-to-date security profiles of `oscap` and you can list the available security profiles for *Ubuntu*

Passing an oscap profile to the *Firefox* browser

Practical application: *You need to obtain a list of security controls to harden Firefox manually*

Open *SCAP Workbench* (*System – Scap workbench*) and use the *File – Open Other Content* option to load a *Firefox* security profile from the folder you used to unzip the contents you downloaded in the previous exercise (`ssg-firefox-ds-1.2.xml` file). Execute *Scan* and click *Show report*, examining the report contents. *What happened? What is the usefulness of this SCAP profile in this operating system? What is the source of this security profile?* (STIG, CIS, other…)

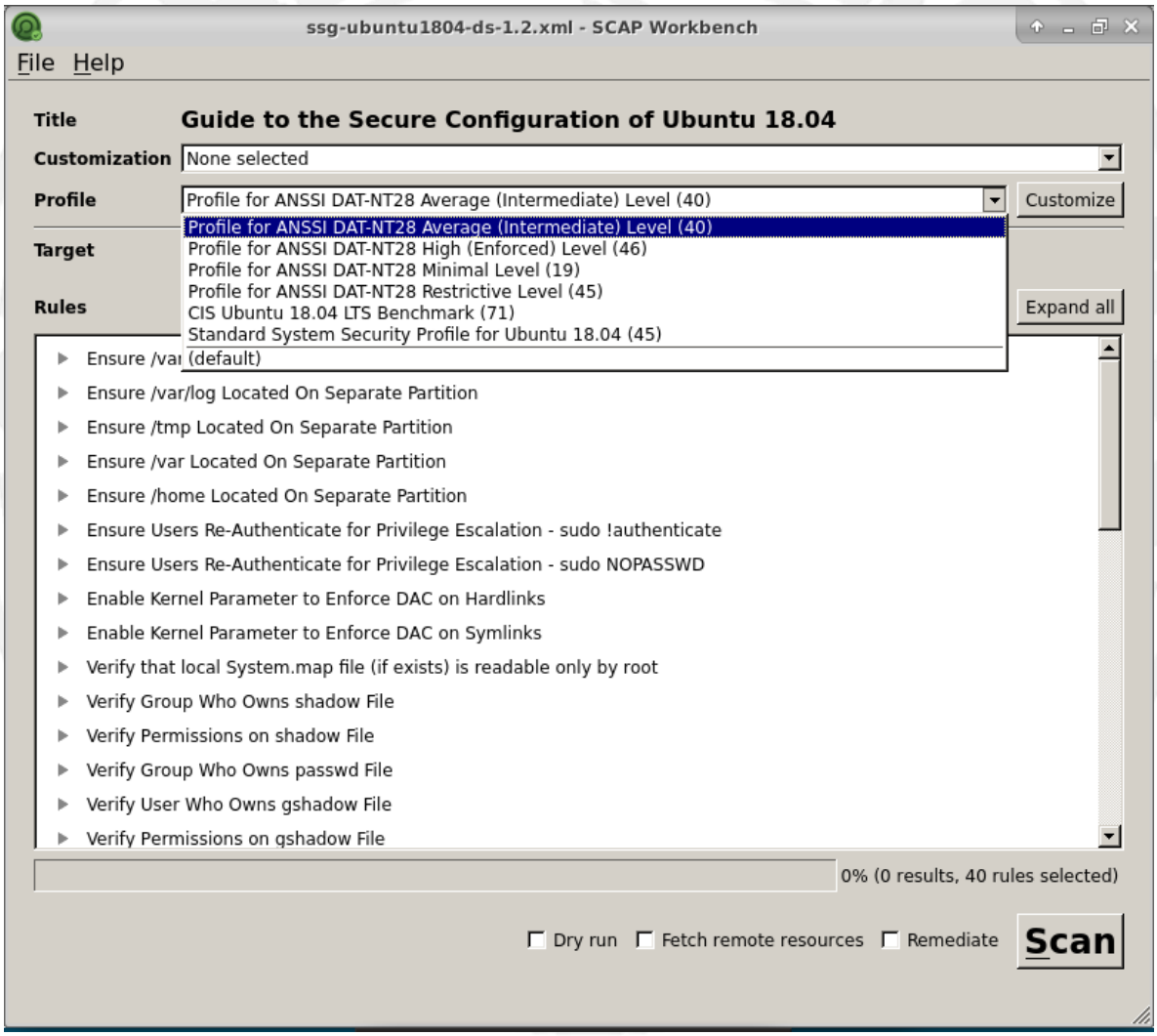
Expected results: This activity will be finished when you run a *Firefox* security profile and are able to interpret the contents of the generated report and its results. Are any of the security controls checked and/or applied?

Passing *OpenSCAP* profiles to the OS

Practical application: *You need to evaluate your Linux OS according to some international security profiles*

Repeat the previous procedure to load the security profile for the operating system of the VM (use the `ssg-ubuntu1804-ds-1.2.xml` file). Analyze the security profiles available and the amount of security controls they contain (between parenthesis). Once loaded, perform the following operations:

- Leave the checkboxes "*Dry Run*" and "*Remediate*" unchecked
- Check the "*Fetch remote resources*" checkbox
- Scan the OS using the following three profiles, **saving each report you obtain after each scan in a different HTML file**. If there are errors reported, ignore them. The profiles are: *ANSSI DAT-NT28 High* profile (a French security standard), the *Ubuntu Standard Security Profile,* and the *CIS Benchmarks* profile.



Expected results: This activity will be finished when you can use *SCAP Workbench* to scan the operating system automatically for security vulnerabilities according to a loaded security profile and store their reports in different files

Interpreting OSCAP reports

Practical application: *You need to obtain an OSCAP report and interpret it*

After running the analysis of the mentioned profiles, open the generated HTML reports and be sure to understand the following report elements.

- The security profile **origin** and initial description
- The **results**: number of rules passed, failed and in other situations
- The **severity distribution** of the failed rules
- The **security score** of the system according to the current security policy
- The classified **security control list** and their pass/fail status
- The **detailed description** of each security control (Description, Rationale, Severity…)

Expected results: This activity will be finished when you are fully able to understand each section of an `oscap` report and locate all the listed elements within it

Using OSCAP from the command-line

Practical application: *You need to run oscap without a GUI*

If we do not have GUI or the possibility to install *SCAP Workbench* you can still perform these operations using the command line. Before continuing this laboratory, you must learn to run a scan like the previous ones using the command line only. To do that consider the following guidelines:

- The command line tool is called `oscap` and requires `root` privileges.
- To eval the security status of the machine it needs two more parameters separated by a blank space: `xccdf eval`. **NOTE**: If we also want to try to automatically remediate the potential security controls that fail the scan (**seen later, do not do that now!**), we must use `xccdf eval --remediate` instead.
- After that, it requires a `--profile` parameter. After a blank space, it requires the fully profile id name to use (with dots) that you can obtain from the `oscap info` command results in the first activity.
- After that, it requires a `--results` parameter. After a blank space, it requires a `.xml` file name to save the report to in this format.
- After that, it requires a `--report` parameter. After a blank space, it requires a `.html` file name to save the "human-readable" report to in this format (this is the one we can see in *SCAP Workbench*).
- Finally, after a blank space, it requires path to the `.xml` security policy file for the current OS to use. These are on the downloaded policy file from *GitHub,* and is the one that defines the security profile specified in `--profile` option (in our case it is the file `ssg-ubuntu1804-ds-1.2.xml`)

Make sure that the report you obtain has the same results than the equivalent run with *SCAP Workbench* (unless you remediated something, in this case the report will probably have more security controls reported as fixed). You only have to test this with one of the three profiles you used in the previous exercise.

Expected results: This activity will be finished when you can use `oscap` from the command line

BLOCK 2. OSCAP REMEDIATION

SCAP Workbench remediation

Practical application: *You need to automatically harden your OS according to the scap-security-guide package policies, using OSCAP remediation*

First, run a *Lynis* audit to obtain a security score and store the result in a file you can consult later. Select the CIS profile again and try to automatically remediate the security controls checking the corresponding "*Remediate*" checkbox. Compare and analyze the results with the original CIS profile verification of the previous section. **Run the *Lynis* audit again and store the report in a different file.** Compare *Lynis* scores and SCAP % of rules correctly applied before and after remediation. Have they improved?

Expected results: This activity will be finished when you use `oscap` or *SCAP Workbench* to attempt to remediate security controls that fails their checks. Once you have the results, you need to compare the *Lynis* scores and SCAP % of correctly applied rules to determine if the remediation had a positive effect, and determine from your results if you think that *Lynis* and SCAP profiles measure different things (in other words, they use different security controls)

Script remediation

Practical application: *You need to automatically harden your OS according to the scap-security-guide package policies, using bash remediation*

Go to the downloaded security profiles and access the `bash` folder inside of it. Locate the shell script corresponding to the *CIS Ubuntu profile*, **make it executable**, and run it with `root` privileges to see how many security controls are remediated now. Validate the security profile again without automatic remediation (via *SCAP Workbench* or the command line) and compare the new report with the previous one. Also, perform a *Lynis* audit again and compare it with the one you did in the previous exercise.

Expected results: This activity will be finished when you use the `bash` remediation scripts included in the security policies file to attempt to remediate security controls that fails their checks. You also need to compare again the % of correctly applied rules and the *Lynis* scores before and after the remediation, do you think now that SCAP profiles and *Lynis* measure different things?

Ansible remediation

Practical application: *You need to automatically harden your OS according to the scap-security-guide package policies, using Ansible remediation*

The final remediation procedure included in the *Oscap* files is through the configuration automation program Ansible. To execute *Ansible* (already installed in the course VM, but installable with : `sudo apt install ansible`) remediation policies you need to perform the following procedure

- Go to the downloaded security profiles, access the `ansible` folder and locate the `.yml` file corresponding to the *CIS Profile*
- Edit this corresponding `.yml` and change `hosts: all` by `hosts: localhost`
- Run `sudo ansible-playbook ubuntu1804-playbook-cis.yml`

- Run a verification of the policy again with `oscap` and a *Lynis* audit to see if the results improve

Expected results: This activity will be finished when you use the *Ansible* remediation scripts included in the security policies file to attempt to remediate security controls that fails their checks. You also need to compare again the % of correctly applied rules and the *Lynis* scores before and after the remediation to determine which remediation procedure obtain better results.

BLOCK 3: EXAMINING STIGS

Using OpenSCAP with STIGs

Practical application: *You need to obtain a security control list to manually harden your OS according to the STIG policies*

The latest STIGs versions can be downloaded from here (https://public.cyber.mil/stigs/downloads/. Search the latest version of an adequate STIG for your *Ubuntu* system (something like "*Canonical Ubuntu 18.04 LTS STIG - Version 2, Release 5*"), download, and unzip it, as we did with the *ComplianceAsCode GitHub* repository contents. Examine its profiles using the `oscap info` commands (or *SCAP Workbench*) and search parallelisms of the profiles you find with the ones in the theory topic. A version of this file is in the laboratory materials if you do not want to download it.

Once the contents are examined, evaluate the current system with any of the available security profiles. Analyze the results and extract conclusions about the potential uses of the provided STIGs for *Ubuntu 18.04* with `oscap`. Can you audit or remediate security controls with this file?

Expected results: This activity will be finished when you can download an adequate STIG profile for the current OS, evaluate it, analyze the results it outputs, and its potential usages

STIG remediation with *Ansible*

Practical application: *You need to automatically harden your Ubuntu OS following the STIG policies*

From the same page of the previous exercise, download the *Ansible* version of the STIG profile (something like "*Canonical Ubuntu 18.04 LTS STIG for Ansible - Ver 2, Rel 5*") and use the same procedure than in the exercise "*Ansible remediation*" to pass the automatic remediations contained on this file. A version of this file is in the laboratory materials if you do not want to download it. Evaluate later the base system with *Oscap* and *Ansible* and compare the results with the previous exercises, determining which automatic remediation procedure is better.

Expected results: This activity will be finished when you can download an adequate STIG *Ansible* remediation script for the previous STIG, run it, and evaluate the final security score of the system.

BLOCK 4: APPLYING A THIRD-PARTY CIS SECURITY POLICY TO AN UBUNTU SERVER

Third-party automated Ubuntu security hardening

Practical application: *You need to automatically harden your Ubuntu OS following the CIS policies using a third-party script provided by Florian Utz*

Florianutz *Ansible* script

*(**NOTE**: After running this remediation script you will lose access to the GUI (having a GUI is a security issue in a server! ☹), so it **is particularly important to have a snapshot created to quickly restore it**)*

Although official automated remediations for CIS policies are not free, there are free (and unofficial) alternatives created by other users and available open source. We are going to use one of them to check the automated hardening capabilities of the CIS policies: https://github.com/florianutz/Ubuntu1804-CIS .

- **We strongly recommend you to create a new VM snapshot/clone to perform this activity**, as the changes performed to the system might give problems. Consult the lab additional documentation to know how to create VM snapshots or clones.
- Now, follow the instructions of the *GitHub* repository of this CIS policy to run it. First, create a `requirements.yml` file with this content (*Ansible* files are processed in *Python*, so please keep the line alignment, and **do not use tabs**). For your convenience, this file is provided as part of the **Lab 6 infrastructure** folder:

```
- src: https://github.com/florianutz/Ubuntu1804-CIS.git
```

- Save this file and install the CIS policy with it using this command

```
sudo ansible-galaxy install -p roles -r requirements.yml
```

*(**NOTE**: If this step fails due to connection problems or unavailability of the repository for some reason, the output of this operation is supplied as part of the lab files. You can obtain the result by unzipping the `CISflorianutz.zip` file we provide in the directory you are working in. This is just a precaution in case this fails, you do not have to do this)*

- Once the hardening policy is installed, prepare a **configuration file** to specify where to run it. To do this, first create the file `server.yml` with the following contents. They specify that the policy will be applied on the own machine (`localhost`), that it requires the user to become `root` to apply it, the name of the roles (set of operations that will be performed, `Ubuntu1804-CIS` as indicated by the policy instructions) to be run, and a descriptive name for the whole operation. For your convenience, this file is provided as part of the **Lab 6 infrastructure** folder.

```
- name: Harden Server
  hosts: localhost
  become: yes

  roles:
    - Ubuntu1804-CIS
```

- Finally, run the policy (`sudo ansible-playbook server.yml`) and wait for the system to be hardened. After that, `reboot`, and login normally before proceeding to the next step to know if the system is operating correctly. The execution process takes a while and looks like this:

```
TASK [Ubuntu1804-CIS : SCORED | 1.1.14 | PATCH | Ensure nodev option set on /dev/shm partition
 SCORED | 1.1.15 | PATCH | Ensure nosuid option set on /dev/shm partition
 SCORED | 1.1.16 | PATCH | Ensure noexec option set on /dev/shm partition] ***
changed: [localhost]

TASK [Ubuntu1804-CIS : NOTSCORED | 1.1.17 | PATCH | Ensure nodev option set on removable media parti
tions] ***
ok: [localhost]

TASK [Ubuntu1804-CIS : NOTSCORED | 1.1.18 | PATCH | Ensure nosuid option set on removable media part
itions] ***
ok: [localhost]

TASK [Ubuntu1804-CIS : NOTSCORED | 1.1.19 | PATCH | Ensure noexec option set on removable media part
itions] ***
ok: [localhost]

TASK [Ubuntu1804-CIS : SCORED | 1.1.20 | PATCH | Ensure sticky bit is set on all world-writable dire
ctories] ***
ok: [localhost]

TASK [Ubuntu1804-CIS : SCORED | 1.1.21 | PATCH | Disable Automounting] ********************************
skipping: [localhost]

TASK [Ubuntu1804-CIS : NOTSCORED | 1.2.1 | PATCH | Ensure package manager repositories are configure
d] ***
ok: [localhost]

TASK [Ubuntu1804-CIS : NOTSCORED | 1.2.2 | PATCH | Ensure GPG keys are configured] ******************
ok: [localhost]

TASK [Ubuntu1804-CIS : SCORED | 1.3.1 | PATCH | Ensure AIDE is installed (install nullmailer instead
 of postfix)] ***
skipping: [localhost]

TASK [Ubuntu1804-CIS : SCORED | 1.3.1 | PATCH | Ensure AIDE is installed] ***************************
```

*(**NOTE**: As we said, this hardening profile uninstalls `xfce4` as it is considered a potential vulnerability. You need to `reboot` the machine immediately after hardening it, as it will hang because of it. You can restore `xfce4` again like this, but we recommend restoring a snapshot / clone to avoid future problems:*

- *Run* `sudo apt install --reinstall xfce4`
- *Reboot and reinstall* `firefox`*)*

After this automated hardening procedure, you should check the CIS profile with *OpenSCAP* again to see if its result changes. Also, you should check the system security with *Lynis* again to see if the index is higher.

Expected results: This activity will be finished when you can harden an *Ubuntu* system using this *Ansible* script to implement a great amount of CIS security controls automatically and evaluate the hardening results.

The Egida Ansible CIS-based automated hardening project

Practical application: *You need to automatically harden your Ubuntu OS following the CIS policies using a third-party script provided by Antonio Payá*

This automated hardening project is similar in nature to the previous one, as it consists of applying CIS benchmarks via *Ansible*. **Please create another snapshot of the machine before running this software**. The advantage it has (**apart from being made in our school** 😊) is that it is designed to be more granular in the future, presents a menu, and a simple configuration to be applied to multiple machines at the same time. For the purpose of our course, simply follow the instructions in the official repository:

https://github.com/Egida-Kassandra/egida/blob/master/docs/index.md#installation for installation and use only on the local machine (`localhost`). If we follow the instructions and run the menu (`sudo egida menu -c local`, and default options) so that all CIS benchmarks are used, we will get a result like this:

```
RUNNING HANDLER [egida-role-cis : sysctl flush ipv4 route table] ***************
changed: [localhost]

RUNNING HANDLER [egida-role-cis : sysctl flush ipv6 route table] ***************
changed: [localhost]

RUNNING HANDLER [egida-role-cis : restart auditd] ******************************
changed: [localhost]

RUNNING HANDLER [egida-role-cis : load audit rules] ****************************
changed: [localhost]

RUNNING HANDLER [egida-role-cis : update grub] *********************************
changed: [localhost]

RUNNING HANDLER [egida-role-cis : restart sshd] ********************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=219  changed=126  unreachable=0    failed=0    s
kipped=20   rescued=0    ignored=1
```

As with the previous benchmark, the machine is left without access to the GUI as it is considered a vulnerability (you can restore it with the same procedure we saw).

After running *Egida*, it is necessary to reevaluate the system again with *OpenSCAP* and *Lynis* to see if the final hardening index has improved. You may also lose internet connectivity in the machine. **Do not try to fix this if it happens, just revert to the previous snapshot**.

Expected results: This activity will be finished when you can harden an *Ubuntu* system using this *Ansible* script to implement a great amount of CIS security controls automatically and evaluate the hardening results, considering the advantages and disadvantages these automated scripts have.

BADGES AND SELF-ASSESSMENT

NOTE: You have a version of this badge table in an editable format available in the *Virtual Campus*. You can use this file to easily create a document in the format you want and take extended notes of your activities to create a log of what you did. Remember that this material is key to keep track of your self-evaluation.

Badge Level	Unlocked when	Unlocked?
	You can check the *OpenSCAP* available security profiles for any supported product	
	You can load an adequate security profile for a supported operating system	
	You can use *OpenSCAP* to attempt to automatically remediate potential problems in the application of security controls. You can answer this question. *Even if not significantly improving the security index of a newly installed machine, what do you think it will happen with a machine which have not been correctly maintained or administered?*	
	You can update the *OpenSCAP* security profile files	
	You can answer the following questions: *What happens if a security control is reported as "not applicable"? what are the potential reasons? Is this security control still useful for securing the system?*	
	You can perform an automated scan of an operating system to detect vulnerabilities according to a certain security profile.	
	You can use `oscap` from the command line to scan and remediate security controls	
	You can answer the following question: *Does the Ubuntu 18.04 STIG include profiles corresponding to all the possible combinations of MAC and confidentiality levels we reviewed in theory?*	
	You can answer the following questions: *Is the Ubuntu 18.04 STIG usable with OpenSCAP to automate verification / remediation of security controls? If your answer is no, what can you do with the information it provides?*	
	You can use the `bash` remediation scripts of any SCAP security policy.	
	You can use the *Ansible* remediation scripts of any SCAP security policy.	
	You can run a STIG remediation procedure with the *Ansible* tool	
	You can fully analyze and understand the contents of an `oscap` report	
	You understand the advantages and disadvantages of automated *Compliance as Code* tools, and how high security scores may have side effects that may prevent machines to work properly and require manual intervention	
	You can answer this question: *which method shows better remediation results in the Ubuntu CIS Profile: Oscap remediation, bash script remediation or Ansible remediation? Is the security improvement significative starting from a newly installed system?*	
	You can run and apply the *Florianutz* third-party automated *CIS*-based *Ansible* hardening script, evaluating the increase in the general hardening index of the system later.	
	You can run and apply the *Egida* third-party automated CIS-based *Ansible* hardening script, evaluating the increase in the general hardening index of the system later.	

	**This is the way**: Your proficiency in automated hardening procedures allows you to secure an *Ubuntu* system according to several security policies that guarantee various levels of security. Also, you can automatically evaluate the security of any *Ubuntu* system. With this, you are prepared to use the same procedures in any other system supporting the same class of automation.	