Source: Bing Chat AI

Laboratory 6. Automatable OS Security and Security Policies

COMPUTER SCIENCE DEPARTMENT. UNIVERSITY OF OVIEDO

Computer Security | 2023 – 2024 (*V4.0 "S-81 Isaac Peral"*)

Content

Infrastructure of this laboratory

This lab **cannot run in containers, as the tools we are going to use are not** designed to run on them. In fact, there is a special version of `oscap` designed to run inside *Docker* containers (`oscap-docker`), but its use is outside the scope of the course. Therefore, **we strongly recommend that you take a snapshot of your current virtual machine** to run the programs in this lab. **See the additional lab documentation to learn how to create snapshots.**

The Lab 6 infrastructure folder contains some files that are needed to perform the activities of this lab, in order to save time.

On the other hand, **you will also need to use the "*Wazuh machine*"** from the previous lab to do a series of checks at the end of the lab. Remember that if you do not have a machine at home capable of supporting the creation of a VM with these characteristics, we recommend that you **prioritize *Wazuh*'s activity to do it in the lab**, where you do have hardware that supports that VM. You can even start with it without any problems.

🔍 ***NOTE: Scap workbench was not available on Ubuntu 22.04 at the time of designing these labs. If you try to install it and it's not possible, it means that it hasn't been added back to your distribution's list of available packages. If this is your case, all you have to do is try to do the exercise on Ubuntu 18.04 or do the same exercises with `oscap`, from the command line. At the end of the day, SCAP Workbench is just a GUI for `oscap` that makes it easy to use, but it doesn't really bring any new functionality.***

Block 1: Managing *OpenSCAP* policies

Exercise L6B1_OSCAPUPD: Updating *OpenSCAP security policies*

Description of the activity / Practical application

You need **to update the hardening rules in the** `scap-security-guide` package to the latest version

Expected results

This activity will end when **you download and install the most up-to-date** `oscap` security profiles and are able to list the security profiles available for *Ubuntu*

Additional information required to do it

The *OpenSCAP Compliance as Code tool* and *SCAP Workbench* (a free *OpenSCAP GUI*) **are already installed in the course's virtual machine.**

If you created your own machine, to install the OpenSCAP tool you need to follow the official installation steps for Ubuntu here: https://www.open-scap.org/tools/openscap-base/#download. Also install SCAP Workbench, with: `sudo apt install scap-workbench`

To ensure the best result when running *OpenSCAP*, **you need to update your security policies**. These contain validation and remediation rules for different operating systems and *software*.

- Download the latest version of these policies from their official *GitHub repository* (https://github.com/ComplianceAsCode/content). Update the version number of this link to the latest one available: `wget https://github.com/ComplianceAsCode/content/releases/download/v0.1.66/scap-security-guide-0.1.66.zip`, or go to the ***Releases*** section of the *GitHub* repository

NOTE: In the files of the Lab 6 infrastructure there is a folder with an old policy of this repository. You can use this file and avoid downloading a more updated one, because for this activity it is not necessary to "be up to date" necessarily

- Unzip the downloaded file and copy its contents to a folder you create. **This folder will be the one you will use later to upload *OpenSCAP* content .**

Once that is done, we will have the latest ***security profiles*** available. Each security profile is tailored to a specific use (organization, public entity, etc.) 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 find out the profiles available for *Ubuntu 22.04* do: `sudo oscap info <unzipped folder>/ssg-ubuntu2204-ds-1.2.xml`

Exercise L6B1_OSCAPAUDITSW: Convert an *OpenSCAP* profile to the *Firefox browser*

Description of the activity / Practical application

You need a list of security controls to harden ***Firefox*** manually

Expected results

This activity will end when **you run a *Firefox* security profile** and are able to interpret the content of the generated report and its results. You can answer this question: *Are any of the security controls checked and/or enforced?*

Additional information required to do it

Open *SCAP Workbench* ***(System – Scap workbench***) and use ***the File – Open Other Content*** option to load a *Firefox* security profile from the folder where you unzipped the file downloaded in the previous exercise (`ssg-firefox-ds-1.2.xml` file). Run ***Scan*** and click ***Show report***, analyzing the contents of the report. *What happened? What is the usefulness of this SCAP profile for this product? What is the origin of this security profile (i.e., where did your technical security controls come from)?* (STIG, CIS, others...)

Exercise L6B1_OSCAPAUDITSO: Passing *OpenSCAP profiles* to the OS

Description of the activity / Practical application

You need to **evaluate your *Linux*** OS according to international security profiles

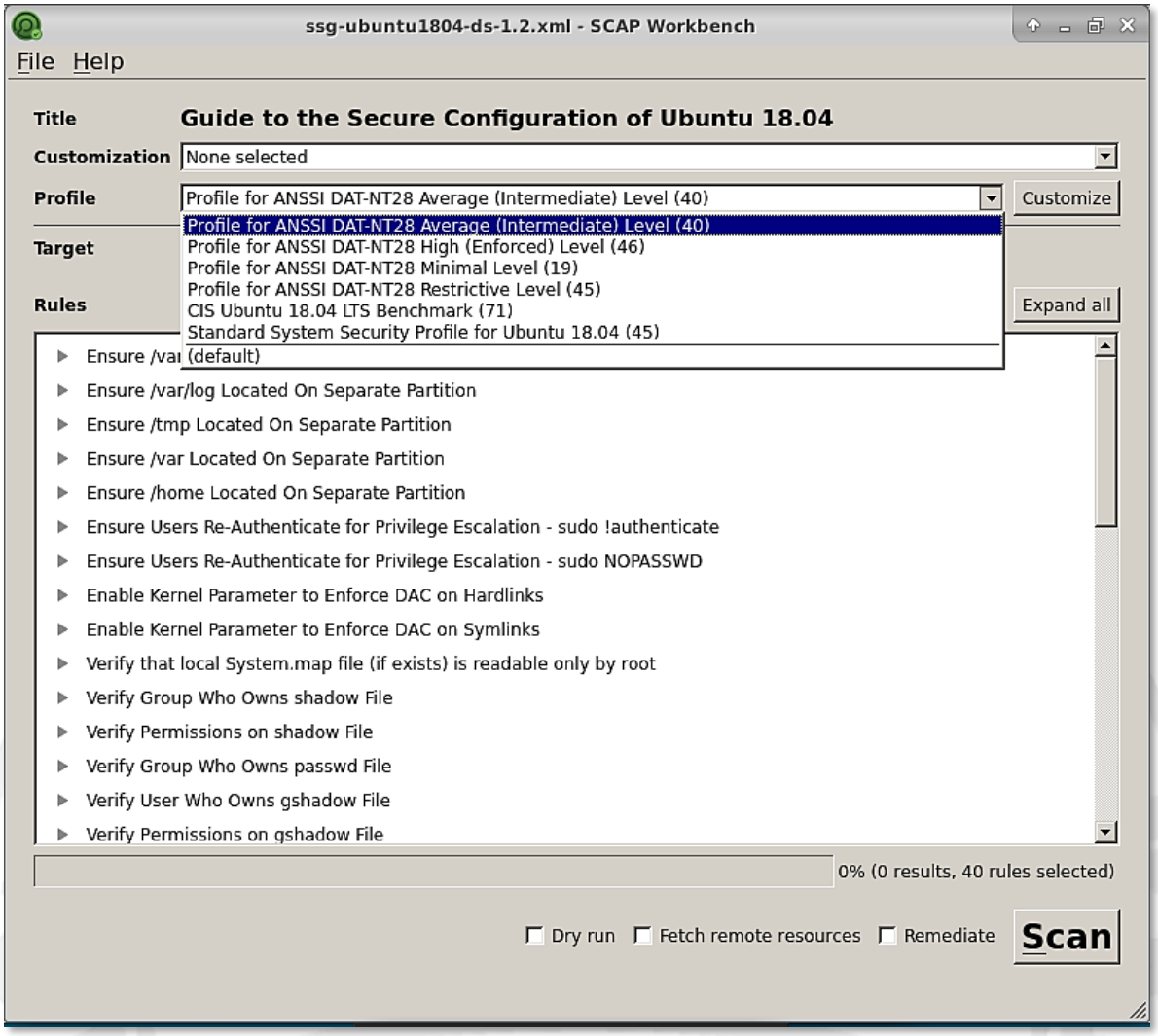
Expected results

This activity will end when you can use *SCAP Workbench* to automatically scan the operating system **for security issues according to a loaded security profile and you can also store its reports in different files**

Additional information required to do it

Repeat the above procedure to load the security profile for your MV's operating system from the folder where you unzipped the rules to download from *GitHub* in the first exercise (use the `ssg-ubuntu2204-ds-1.2.xml` file). Analyzes the available security profiles and the number of security controls they contain (in parentheses). Once charged, do the following:

- Leave "***Dry run"*** and **"Remediate**" unchecked
- Brand *"**Fetch remote resources***"
- Scan the operating system using the following three profiles, **saving each report you get after each scan in a different HTML file.** If errors are reported, ignore them. The profiles are:
  - *ANSSI DAT-NT28 High profile* (a French security standard)
  - The *Ubuntu* standard security profile
  - The profile that follows the *CIS Benchmarks*.



Exercise L6B1_OSCAPREPORT: Interpreting *OpenSCAP reports*

Description of the activity / Practical application

*You need **to get an OpenSCAP security report** and know how to interpret it*

Expected results

This activity will end when **you understand each section of an** `oscap` report and locate all of the items listed within it.

Additional information required to do it

After running the analysis of the profiles mentioned in the previous exercise, open the generated HTML reports, and make sure you understand the following elements of the report.

- The **source** of the security profile and initial description.
- The **results:** number of rules passed, rules that the system does not comply with, and in other situations
- The **severity distribution** of rules that the system does not comply with
- The **system's security score** based on the security policy used

- The **list of classified security** controls and their current status of passed and unpassed controls
- Detailed **description** of each security check (Description, Justification, Severity...)

Exercise L6B1_OSCAPCMD: Using OSCAP from the command line

Description of the activity / Practical application

You need to run `oscap` without a *GUI*

Expected results

This activity will end when you are able **to use** `oscap` **from the command line**. You can answer this question: *Do you understand when you will have no choice but to run OpenSCAP like this?*

Additional information required to do it

One thing we must keep in mind is that in most cases server machines will not install a GUI to save resources and reduce attack vectors. If we do not have a GUI, or the ability to install *SCAP Workbench,* we can perform the operations of the previous exercises **using the command line**. Before proceeding with this lab, you should learn how to run a scan like the ones above using only the command line. To do this, use the following information:

- The command-line tool is called `oscap` and requires `root` privileges.
- To assess the security status of the machine, you need to add two parameters to the previous command separated by a blank space: `xccdf eval`. **NOTE:** If we also want to try to automatically remediate possible security controls that fail the scan (**see below, do not do it now!**), we should use `xccdf eval --remediate` instead.
- After that, it requires a `--profile` parameter. After a blank space, you need to put the full profile id name (with dots) that you can get from the results of the `oscap info` command of the first activity.
- After the profile, a `--results` parameter is required. After a blank space, you need an `.xml` file name to save the report in this format.
- After the results, a `--report` parameter is required. After a blank space, it has a `.html` file name to save a "human-readable" report in this format (this is the one we can see in *SCAP Workbench).*
- Finally, after a blank, you need the path to the `.xml` file with the security policy to use with the current operating system. These files are in the policy file downloaded *from GitHub* that we unzip, and it is the one that defines the name of the security profile specified in the `--`profile option (in our case it is the `ssg-ubuntu1804-ds-1.2.xml` file)

Make sure that the report you get has the same results as the equivalent run with SCAP Workbench (unless you've fixed something before, in which case the report will likely have more security controls fixed). Just try this with one of the three profiles you used in the previous exercise.

Block 2. Remediation with *OpenSCAP*

Exercise L6B2_OSCAPREMWB: Remediation with *SCAP Workbench*

Description of the activity / Practical application

You need to **automatically harden your OS** by following the policies in the `scap-security-guide` package using OSCAP remediation

Expected results

This activity will end when you use `oscap` or *SCAP Workbench* to attempt to **fix security controls that fail their checks**. Once you have the results, you should **compare the *Lynis* score *and the % of correctly applied* SCAP rules to determine if the fix had a positive effect**, and determine from your results whether you think the *Lynis* and SCAP profiles measure different things (in other words, if they use different security controls)

Additional information required to do it

First, run an audit with *Lynis* to get a baseline security score and store the result in a file that you can refer to later. Reselect the CIS profile and attempt to automatically fix its security controls by checking "*Remediate*" in the GUI. Compare and analyze the results with the initial execution of the CIS profile in the previous section. **Run the *Lynis* audit again and store the report in a different file.** Compares the *Lynis* score and the percentage of SCAP rules applied correctly before and after the correction. *Have they improved?*

Exercise L6B2_OSCAPREMBASH: Remediation with *Bash* scripts

Description of the activity / Practical application

You need to ***automatically*** **harden** your OS by following the policies in the `scap-security-guide` package **using *bash* scripted *remediation***

Expected results

This activity will end when **you use the `bash` remediation scripts** included in the downloaded security policy file to attempt to remediate security controls that fail their checks. You should also compare again the % of correctly applied rules and the *Lynis* scores before and after remediation. *Do you think now that the SCAP and Lynis profiles measure different things?*

Additional information required to do it

Go to the downloaded and unzipped security profiles in a folder at the beginning of the lab and access the `bash` folder within it. Locate the *shell script* corresponding to the *Ubuntu* CIS security profile, ***make it executable,*** and run it with `root` privileges to see how many security controls have now been fixed. Audit the OS with the security profile again **without automatic remediation** (via *SCAP Workbench* or command

line) and compare the new report with the old one. Also, do a *Lynis audit again* and compare it to the one you did in the previous exercise.

Exercise L6B2_OSCAPREMANS: Remediation with *Ansible*

Description of the activity / Practical application

You need to automatically ***harden* your OS** by following the policies in the `scap-security-guide` package using remediation **with *Ansible***

Expected results

This activity will end when **you use the *Ansible* remediation *scripts*** included in the downloaded security policy file to attempt to remediate security controls that fail your checks. You should also compare the % of correctly applied rules and the *Lynis* scores before and after remediation again to determine which remediation procedure gets the best results.

Additional information required to do it

The final remediation procedure included in the `oscap` files is through the ***Ansible*** configuration automation program. To run the remediation policies *with Ansible* (already installed on the virtual machine of the course, but installable with: `sudo apt install ansible`), you must do 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 `.yml` and change `hosts: all` to `hosts: localhost`
- Run `sudo ansible-playbook ubuntu1804-playbook-cis.yml`
- Run a policy check again with `oscap` and an audit with *Lynis* to see if the results improve

Block 3: Audit and remediation with STIGs

Exercise L6B3_STIGSCCAUDIT: Automated audit with *SCAP Compliance Checker* (SCC) and STIGs

Description of the activity / Practical application

It consists of using the *SCAP Compliance Checker (*SCC) program and the STIGs that come with it **to audit an *Ubuntu* system according to the specifications of the STIGs of the DISA** (*Defense Information Systems Agency*) of the United States.

Expected results

You can use the aforementioned tool to **audit your *Ubuntu* VM and generate a report with its results**, understanding the process and the information that the tool returns. You can answer these questions, in addition to those posed in the exercise:

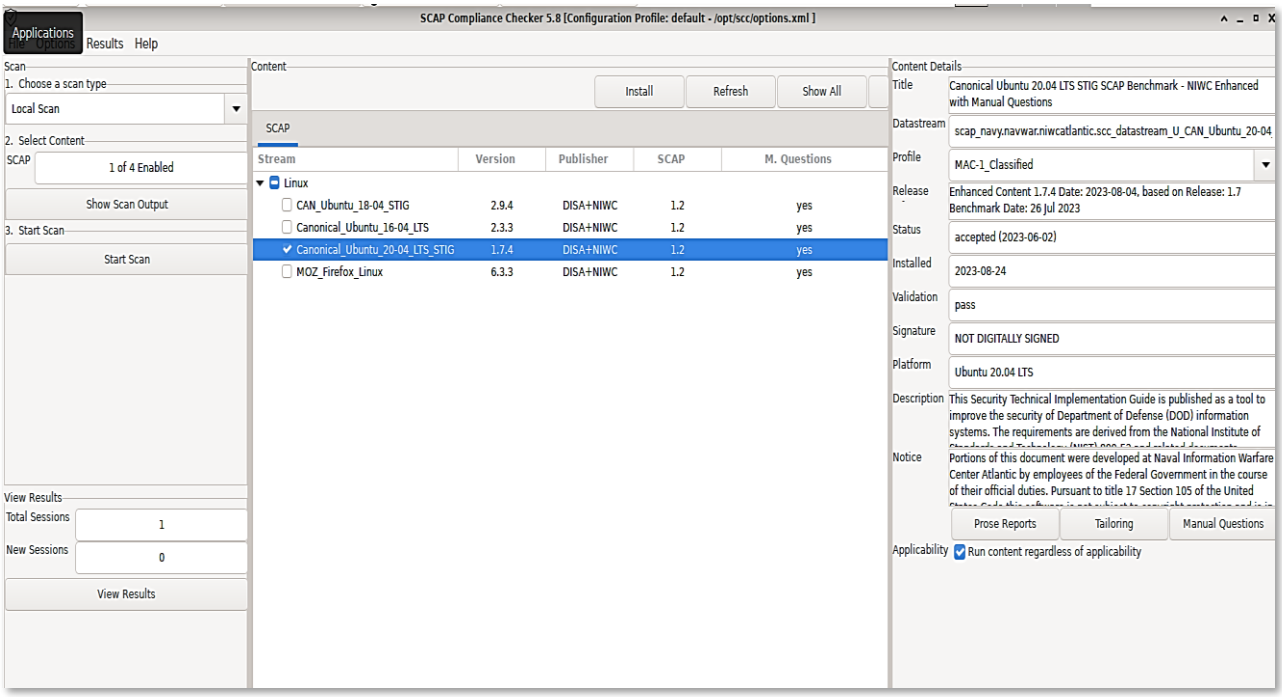
- *Do you locate in the tool the MAC levels and other elements defined for the STIGs in the theory, so that you can choose under which conditions you pass the audit?*
- *Do you realize that the generated report not only gives you an evaluation, but details everything you have to do to improve the controls that the system does not pass?*
- *Do you understand that this tool does automatic auditing, but not remediation?*

Additional information required to do it

If you visit https://public.cyber.mil/stigs/scap/ and search through the **SCAP 1.2** content, you will see that very recently the US has made **an official SCAP automation tool generally available for some products that, until now, only allowed to use to its government suppliers**. We mentioned this tool in the theory, and it is called **SCC** (*SCAP Compliance Checker*). We can download the tool on our *Ubuntu 22.04 VM* using this link: https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/scc-5.8_ubuntu22_amd64_bundle.zip

Download the tool and unzip it. In it you will see a `.deb` file that has the installation of this tool. In a *GNOME* GUI it would be enough to double-click on that file to start installing it, but in XFCE4 we have to do it via console with: `sudo dpkg -i <path of the file .deb>`

Once the tool is installed, you will be able to run it with `sudo /opt/scc/scc` and you will see its interface (ignore if there are any warnings).



As you can see, by default the tool includes 4 STIGs, three of them corresponding to past versions of ***Ubuntu*** and one of them to ***Firefox***. Before use, please click the ***Refresh button***.

> 🔍 ***It is possible that if you download it now, the Ubuntu 22.04 STIG will already appear. If so, use it instead of the one we are going to mention now.***

To test the tool, **check only the *Ubuntu* 20** STIG, even if it is not the correct version of your OS, and make sure that the "***Run content regardless of applicability***" option is checked so that SCC will scan the OS as well. Locate the place where the scan profiles can be set up and see that all the combinations seen in the theory are there.

> 🔍 ***For this test, simply select a MAC 1 Classified.***

When you launch the scan on the left button, the system warns you that that STIG has some controls that **ask you questions that you have to answer manually,** and how to create the answers. We are not going to do that in this test, so we can just ***Continue*** to scan the system with everything that is automatable.

Wait for the system audit to finish and a report to be generated. To be able to see it from the tool you need to install a separate tool, but it is not necessary in our case. Simply search the home ***directory* of the current user** and open the HTML file that you will find in `SCC/Sessions/<date and time when the report was released>`. Within the report, go to "***All Settings***" and answer the following questions:

- *Is there an overall compliance score for the system?*
- *Is a newly installed Ubuntu close to achieving compliance with the DISA STIG or rather far away?*
- *Are the results categorized by its CAT clearly and easily accessible in the form of a list of problems to be solved?*
- *Does each result have enough detail to know how to solve the problem they describe within the report itself?*

> 🔍 ***DISA does publish content for automatic remediation, but not with the characteristics of the SCAP protocol, but with Ansible, Chef (a tool that serves the same purpose as Ansible, deploying configurations in***

an automated way) or directly Powershell. You can find the ones that are publicly available here: https://public.cyber.mil/stigs/supplemental-automation-content/.

Unfortunately, there is not one for all systems covered by STIGs, only for a few (for example, for Ubuntu 22 none are available yet, at least in December 2023)

Exercise L6B3_STIGSCCREM: Example of remediation with *SCAP Compliance Checker* (SCC) and STIGs for *Firefox*

Description of the activity / Practical application

It consists of **learning more about the SCC options** with the profile you have for *Firefox*, including **updating the tool with new STIGs** and learning how to pass it again when a problem is remedied to see the results.

Expected results

You can **evaluate *Firefox* with SCC and learn how to improve its security** by following the instructions on your STIG DISA to improve its overall security. You can answer these questions:

- *Do you understand why so much emphasis on "shielding" a browser?*
- *Do you understand that Firefox security can be improved with the options you put in a single policy file?*
- *Do you think this makes it easier to automatically deploy Firefox Secures?*

Additional information required to do it

SCC is a program that **supports new STIGs** that are released over time or that do not come by default with the tool (such as one of *Ubuntu 22* that the DISA will publish in the future). To find out what is available, you need to go back to https://public.cyber.mil/stigs/scap/ and search through the SCAP 1.2 content. Do so and **find the latest version of *Firefox* for *Linux* STIG available**. Download its zip to your virtual machine.

Once you have that STIG, go to SCC, and use the "***Install***" button above the current list of STIGs. Now all you have to do is search for the zip you downloaded and select it for the tool to add it to its list. Notice that in this case it is a more updated version than the one that comes by default.

Make sure that only the **new *Firefox* STIG is checked** and, before launching it as in the previous exercise, **select MAC 1 Classified *as the profile to be evaluated*** and that it runs it anyway even if it is not applicable. *What is the result?*

Since the scanning results can be clearly improved, we are going to learn how to **secure *Firefox*** in this environment so that, if you want, you can make a security policy that you can then apply whenever you want. It turns out that when you start *Firefox*, the program is set up to look for a file called `policies.json` in a predetermined path. In the case of *Ubuntu,* it's `/etc/firefox/policies`. If you create that folder, you can create that file inside it by following these instructions: https://support.mozilla.org/en-US/kb/customizing-firefox-using-policiesjson.

Now, look at the report that SCC has generated **and implement some of the solutions suggested by the STIG** that involve changing that file, and adding a directive and a value following that report. Once you have incorporated a couple of them, **re-evaluate *Firefox*** and answer these questions:

- *Is SCC looking at your Firefox settings or is it just looking at the contents of this policy file and nothing else?*
- *What would you do if you got a* `policies.json` *file that returned a very high score and had to create a VM to clone later that allowed safe browsing?*

Block 4: Audit and remediation with *CIS Benchmarks*

Exercise L6B4_CISUBUNTUPRO: The *Ubuntu Pro license*

Description of the activity / Practical application

It consists of taking advantage of the functionalities of *Ubuntu Pro when* it comes to **auditing and automated remediation of CIS security controls** to an *Ubuntu* system.

Expected results

You can use *Ubuntu Pro's* capabilities to audit and remediate CIS-recommended security controls for an Ubuntu system, **using a profile appropriate for the audited system**. You can answer the questions asked in the exercise and also these:

- *Do you think that the Ubuntu Pro license makes hardening an Ubuntu system very easy?*
- *Does the Ubuntu system lose its GUI when hardening?*
- *After examining the documentation, would you know how to do an equivalent process to harden Ubuntu using the DISA (Defense Information Systems Agency) STIG security controls? What do you think would happen now if we passed the SCC from the previous block once the corresponding fix process was done? (it is not about doing it, but about knowing how it would be done and its consequences)*
- *Also, would you know how to generate a fix for an Ubuntu system that does not have the necessary packages installed to do this exercise with the tools provided? (it is not about doing it; it is about knowing how it would be done)*

Additional information required to do it

To do this exercise **you need to create a personal account in *Ubuntu Pro*** (it is free): https://ubuntu.com/pro . Once you do it and confirm your email (you can give Uniovi's) you can log in to your account and in ***"Your Subscriptions***" locate your "***Free Personal Token***". This is the token you need to follow.

🔍 ***Ubuntu Pro is free for up to 5 machines. If you are worried about "burning" one of your 5 uses in this test, do not worry. They are 5 active machines. If a machine does not contact the Ubuntu Pro server within 24 hours, it will be automatically removed from your list of active machines, so you will not lose one of your uses*** 😊

To do this exercise, follow these steps:

- Just in case, **do not forget to take a *snapshot*** or linked clone of the VM so you can revert to a base state if something goes wrong.
- **Audit the base system with `lynis`** and save the generated report, just like in **Lab 1**.
- **Install the *Ubuntu Advantage*** (UA) client and attach your subscription token as follows: https://ubuntu.com/security/certifications/docs/disa-stig/installation

- **Run a full CIS audit process** (without remediating security controls) by following these instructions: https://ubuntu.com/security/certifications/docs/usg/cis/audit. Use the **appropriate profile,** keeping in mind that we want **controls that don't affect the functionality of *Ubuntu*** and that the VM is a ***Server***: https://ubuntu.com/security/certifications/docs/usg/cis/compliance. Answer these questions:
  - *Is a base Ubuntu VM closer to having an acceptable % compliance with the CIS policies and with the DISA STIGs of the previous block?*
  - Based on that, *which of the two do you think is stricter in terms of the level of security they impose?*
- Wait for the audit to finish and **analyze the SCAP report** generated in `/var/lib/usg/`, which has the same format as the one we saw with *OpenSCAP* in block 1. Keep in mind that the report is generated with `root` permissions and in a protected folder, so you will need to change the owner file to be able to read it without problems. *Is the system far from having acceptable compliance with CIS controls?*
- Run the `fix` procedure by following these instructions: https://ubuntu.com/security/certifications/docs/usg/cis/compliance. Answer these questions now:
  - *Do you have internet connectivity?* If the answer is no, *do you understand why automatic remediation procedures should be run with caution?*

🔍 ***In this case, one of the fixes causes the DNS to be misconfigured, but it can be fixed like this: https://www.ionos.com/digitalguide/server/configuration/change-dns-server-on-ubuntu/. You can put Google's, NextDNS from lab 1, or any other you want*** 😊

  - Many of the *third-party scripts that apply CIS controls* **uninstall the GUI** of an *Ubuntu Server*, either by mistake or because they only implement the strictest profiles. *Has this happened to you with the features of Ubuntu Pro? Why do you think that was?*
- **Go through an audit process again** and see the generated report one more time. *Is the system far from having acceptable compliance with CIS controls?*

Exercise L6B4_CISWAZUH: *Wazuh* as CIS policy manager

Description of the activity / Practical application

It consists of using the *Wazuh* installation that we have done in the previous laboratory to start **the automated evaluation of the monitored systems with CIS policies** linked to them.

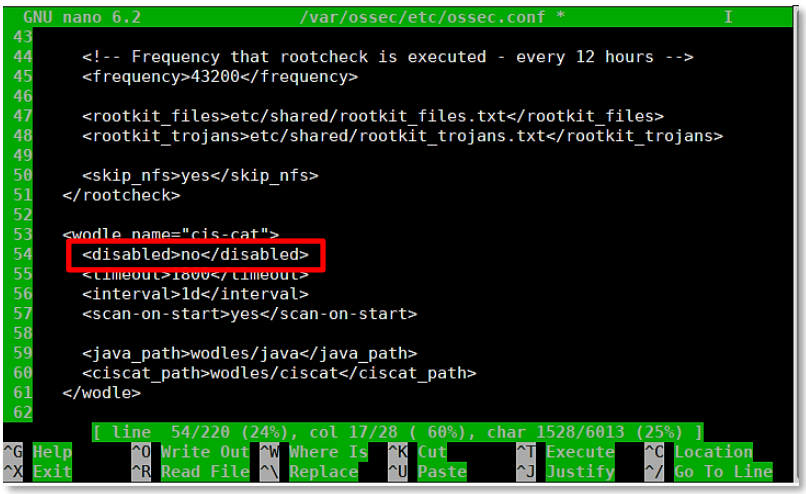
Expected results

You can make ***Wazuh* evaluate the system it monitors** with the CIS policies associated with it, and **thus obtain a CIS security index**, which security controls it passes and which it does not. You can answer these questions:

- *Does Wazuh consider that, if the monitored system has a very low score, it is a security event that needs to be reported?*
- *Can you get a list of the security controls that the system does not pass and all the details of each control (audit, remediation, description...), equivalent to what the corresponding CIS Benchmark PDF has?*

Additional information required to do it

One of the great advantages of using *Wazuh* is that it comes with an entire **automated audit management system with *CIS Benchmarks* built in**. However, in order to be able to use it in the most effective way, it is advisable **to modify the OSSEC configuration file** of the agent in the monitored system to activate the integration with the **CIS-CAT** evaluation tool mentioned in the theory. Do not forget to restart the agent after making the change.



If everything is correct, you should now be able to click on the agent within *Wazuh* and use the SCA (***Security Configuration Assessment***) option. Look inside it for the system's report made, its score, and if you can find all the details of the controls that the monitored system does not pass.

✍ Badges & Self-Assessment

NOTE: You have a version of this badge table in editable format available on the *Virtual Campus*. You can use this file to create a document in any format you want and take extended notes of your activities to create a log of what you have done. Remember that the material you make can be taken for laboratory tests.

Badge Level	Unlocked when	Unlocked?
	You can use the available *OpenSCAP* security profiles for any supported product	
	You can upload a security profile suitable for a supported operating system	
	You can use *OpenSCAP* to automatically try to troubleshoot potential issues in the application of security controls. You can answer this question. *Even if it does not significantly improve the security index of a newly installed machine, what do you think would happen to a machine that has not been properly maintained or managed?*	
	You can use *OpenSCAP security profile files*	
	You can answer the following questions: *What happens if a security control is listed in a report as "not applicable"? What are the possible reasons? Is this security control still useful in improving system security?*	
	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 analyze and remediate security controls	
	You can answer the following question: *Do the DISA STIGs include profiles corresponding to all the possible combinations of MACs and sensitivity levels that we saw in theory?*	
	You can answer the following questions: *Is SCC for auditing and remediation or just auditing? How could you improve security if it is just an audit?*	
	You can use the bash remediation scripts of any SCAP security policy.	
	You can use the *Ansible* remediation *scripts* for any SCAP security policy.	
	You can create a remediation procedure for *Firefox* by following the DISA STIG security policy	
	You can analyze and understand the content of an `oscap` report	
	You understand the advantages and disadvantages of *Compliance as Code* tools, and how trying to achieve high security scores in this way can have side effects that can prevent machines from working properly and require manual intervention	
	You can answer this question: *which automated method has better remediation results in the Ubuntu CIS profile in OpenSCAP: Oscap fix, bash*	

	script fix, or Ansible fix? Is the security improvement significant from a newly installed system?	
	You understand how to take advantage of the security benefits of *the Ubuntu Pro* license and how to automate the remediation of security issues	
	You understand how *Wazuh* can be used to know very precisely the security level of each of the machines it has audited following the CIS profiles	
	**This is the way**: Your mastery of automated *hardening* procedures allows you to improve the security of an *Ubuntu* system according to various security policies that guarantee different levels of security. In addition, you can automatically assess the security of any *Ubuntu system* or other *software* with world-renowned professional tools. With this, you are prepared to use the same procedures in any other system that supports the same kind of automation.	