
*Fuente: Bing Chat AI*

# Laboratorio 6. Seguridad de SO Automatizable y Políticas de Seguridad

**DEPARTMENTO DE INFORMÁTICA. UNIVERSIDAD DE OVIEDO**

**Seguridad de Sistemas Informáticos | 2023 – 2024** (V4.0 *"S-81 Isaac Peral"*)

# Contenido

# Infraestructura de este laboratorio

Este laboratorio **no se puede ejecutar en contenedores,** ya que las herramientas que vamos a utilizar no están diseñadas para ejecutarse en ellos. De hecho, existe una versión especial de `oscap` diseñada para ser ejecutada dentro de contenedores *Docker* (`oscap-docker`), pero su uso está fuera del alcance de la asignatura. Por lo tanto, **te recomendamos encarecidamente que hagas una instantánea de tu máquina virtual actual** para ejecutar los programas de este laboratorio. **Consulta la documentación de laboratorios adicional** para saber **cómo crear instantáneas.**

**La carpeta de infraestructura del Lab 6** contiene algunos archivos necesarios para realizar las actividades de este laboratorio, para ahorrar tiempo.

Por otro lado, también **tendrás que usar la "máquina *Wazuh*"** del laboratorio anterior para hacer una serie de comprobaciones al final del laboratorio. Recuerda que si no tienes en casa una máquina capaz de soportar la creación de una MV con estas características, te recomendamos que le **des prioridad a la actividad de *Wazuh* para hacerla en el laboratorio**, donde sí tienes hardware que soporte esa MV. Puedes incluso empezar por ella sin problemas.

🔍 ***NOTA: Scap workbench no estaba disponible en Ubuntu 22.04 en el momento de diseñar estos laboratorios. Si intentas instalarlo y no es posible, significa que no se ha vuelto a incorporar a la lista de paquetes disponible de tu distribución. Si es tu caso, solo te queda intentar hacer el ejercicio en Ubuntu 18.04 o bien hacer los mismos ejercicios con `oscap`, desde la línea de comandos. Al fin y al cabo, SCAP Workbench solo es un GUI para `oscap` que facilita su manejo, pero no aporta ninguna funcionalidad nueva realmente.***

# Bloque 1: Administración de directivas *OpenSCAP*

## Ejercicio L6B1_OSCAPUPD: Actualizando las políticas de seguridad de *OpenSCAP*

### Descripción de la actividad / Aplicación práctica

Necesitas **actualizar las reglas de hardening** del paquete `scap-security-guide` a la última versión

### Resultados Esperados

Esta actividad finalizará cuando **descargues e instales los perfiles de seguridad más actualizados** de `oscap` y puedas enumerar los perfiles de seguridad disponibles para *Ubuntu*

### Otra información necesaria para su realización

La herramienta de *Compliance as Code OpenSCAP* y *SCAP Workbench* (una GUI gratuita de *OpenSCAP)* **ya están instaladas en la máquina virtual de la asignatura.**

🔍 ***Si creaste tu propia máquina, para instalar la herramienta OpenSCAP tienes que seguir los pasos de instalación oficiales para Ubuntu que aparecen aquí: https://www.open-scap.org/tools/openscap-base/#download. Instala también SCAP Workbench, con:*** `sudo apt install scap-workbench`

Para garantizar el mejor resultado al ejecutar *OpenSCAP* **es necesario actualizar sus políticas de seguridad**. Estas contienen reglas de validación y corrección para diferentes sistemas operativos y *software*.

- Descarga la última versión de estas políticas desde su repositorio oficial de *GitHub* (https://github.com/ComplianceAsCode/content). Actualiza el número de versión de este enlace al último que esté disponible: `wget https://github.com/ComplianceAsCode/content/releases/download/v0.1.66/scap-security-guide-0.1.66.zip`, o vete a la sección ***Releases*** del repositorio de *GitHub*

🔍 ***NOTA: En los ficheros de la infraestructura del Lab 6 hay una carpeta con una política antigua de este repositorio. Puedes usar este fichero y evitar descargar uno más actualizado, porque para esta actividad no hace falta "estar a la última" necesariamente***

- Descomprime el archivo descargado y copia su contenido en una carpeta que crees. **Esta carpeta será la que usarás más adelante para cargar los contenidos de *OpenSCAP*.**

Una vez hecho esto, tendremos los últimos ***perfiles de seguridad*** disponibles. Cada perfil de seguridad se adapta a un uso específico (organización, entidad pública...) e incluye una lista diferente de controles de seguridad. Al seleccionar uno de ellos, puedes verificar **cuántas de sus reglas de seguridad cumple tu**

**sistema** e incluso corregir automáticamente algunas de ellas. Para saber los perfiles disponibles para *Ubuntu 22.04* haz: `sudo oscap info <carpeta descomprimida>/ssg-ubuntu2204-ds-1.2.xml`

# Ejercicio L6B1_OSCAPAUDITSW: Pasar un perfil *OpenSCAP* al navegador *Firefox*

## Descripción de la actividad / Aplicación práctica

Necesitas una lista de controles de seguridad para hacer ***hardening* de *Firefox*** manualmente

## Resultados Esperados

Esta actividad finalizará cuando **ejecutes un perfil de seguridad de *Firefox*** y seas capaz de interpretar el contenido del informe generado y sus resultados. Puedes contestar a esta pregunta: *¿Se comprueba y/o aplica alguno de los controles de seguridad?*

## Otra información necesaria para su realización

Abre ***SCAP Workbench*** (***System – Scap workbench***) y usa la opción ***File – Open Other Content*** para cargar un perfil de seguridad de *Firefox* **desde la carpeta en la que descomprimiste el fichero descargado en el ejercicio anterior** (fichero `ssg-firefox-ds-1.2.xml`). Ejecuta ***Scan*** y haz clic en ***Show report***, analizando el contenido del informe. *¿Qué ha pasado? ¿Cuál es la utilidad de este perfil SCAP para este producto? ¿Cuál es el origen de este perfil de seguridad (es decir, de dónde se han sacado sus controles técnicas de seguridad)?* (STIG, CIS, otros...)

# Ejercicio L6B1_OSCAPAUDITSO: Pasar perfiles de *OpenSCAP* al SO

## Descripción de la actividad / Aplicación práctica

Necesitas **evaluar tu SO *Linux*** de acuerdo con perfiles de seguridad internacionales

## Resultados Esperados

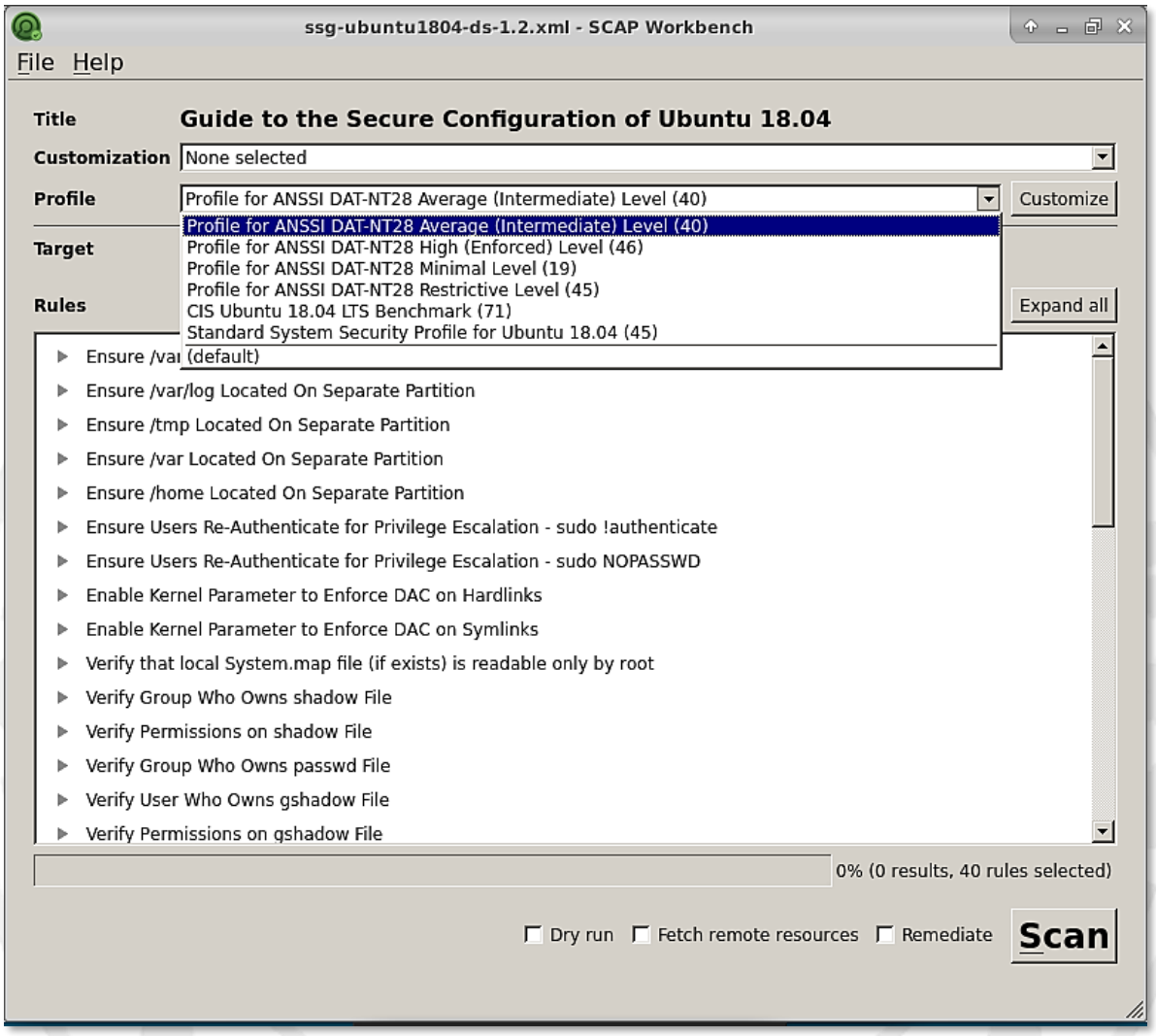
Esta actividad finalizará cuando puedas usar *SCAP Workbench* para **escanear el sistema operativo automáticamente** en busca de problemas de seguridad de acuerdo con un perfil de seguridad cargado y puedas también almacenar sus informes en diferentes archivos

## Otra información necesaria para su realización

Repite el procedimiento anterior para cargar el perfil de seguridad para el sistema operativo de tu MV de la carpeta donde descomprimiste las reglas que descargarse de *GitHub* en el primer ejercicio (usa el archivo `ssg-ubuntu2204-ds-1.2.xml`). Analiza los perfiles de seguridad disponibles y la cantidad de controles de seguridad que contienen (entre paréntesis). Una vez cargados, haz las siguientes operaciones:

- Deja "***Dry run***" y "***Remediate***" sin marcar
- Marca "***Fetch remote resources***"

- Escanea el sistema operativo utilizando los siguientes tres perfiles, **guardando cada informe que obtengas después de cada escaneo en un fichero HTML diferente.** Si se notifican errores, ignóralos. Los perfiles son:
  - *ANSSI DAT-NT28 High profile* (un estándar de seguridad francés)
  - El perfil de seguridad estándar de *Ubuntu*
  - El perfil que sigue los *CIS Benchmarks*.




# Ejercicio L6B1_OSCAPREPORT: Interpretación de informes *OpenSCAP*

## Descripción de la actividad / Aplicación práctica

*Necesitas **obtener un informe de seguridad OpenSCAP** y saber interpretarlo*

## Resultados Esperados

Esta actividad finalizará cuando **comprendas cada sección de un informe** de `oscap` y localices todos los elementos enumerados dentro de él.

### Otra información necesaria para su realización

Después de ejecutar el análisis de los perfiles mencionados en el ejercicio anterior, abre los informes HTML generados y asegúrate de comprender los siguientes elementos del informe.

- El **origen** del perfil de seguridad y la descripción inicial.
- Los **resultados:** número de reglas pasadas, reglas que el sistema no cumple y en otras situaciones
- La **distribución de gravedad** de las reglas que el sistema no cumple
- La **puntuación de seguridad** del sistema según la política de seguridad usada
- La **lista de controles de seguridad** clasificada y su estado actual de controles pasados y no pasados
- La **descripción detallada** de cada control de seguridad (Descripción, Justificación, Gravedad...)

## Ejercicio L6B1_OSCAPCMD: Uso de OSCAP desde la línea de comandos

### Descripción de la actividad / Aplicación práctica

*Necesitas ejecutar* `oscap` *sin un GUI*

### Resultados Esperados

Esta actividad finalizará cuando puedas **usar** `oscap` **desde línea de comandos**. Puedes contestar a esta pregunta: *¿Entiendes en qué casos no te va a quedar más remedio que ejecutar OpenSCAP así?*

### Otra información necesaria para su realización

Una cosa que debemos tener en cuenta es que en la mayoría de las ocasiones las máquinas servidoras no instalarán un GUI para ahorrar recursos y disminuir vectores de ataque. Si no tenemos GUI, o la posibilidad de instalar *SCAP Workbench,* podemos realizar las operaciones de los ejercicios anteriores **usando la línea de comandos**. Antes de continuar con este laboratorio, debes aprender a ejecutar un escaneo como los anteriores utilizando solo la línea de comandos. Para ello, usa la siguiente información:

- La herramienta de línea de comandos se llama `oscap` y requiere privilegios de `root`.
- Para evaluar el estado de seguridad de la máquina se necesita añadir dos parámetros al comando anterior separados por un espacio en blanco: `xccdf eval`. **NOTA:** Si también queremos intentar remediar automáticamente los posibles controles de seguridad que fallan en el escaneo (**visto más adelante, ¡no lo hagas ahora!**), debemos usar `xccdf eval --remediate` en su lugar.
- Después de eso, requiere un parámetro `--profile`. Después de un espacio en blanco, hay que poner el nombre de id de perfil completo (con puntos) que puedes obtener de los resultados del comando `oscap info` de la primera actividad.
- Tras el perfil es necesario un parámetro `--results`. Tras un espacio en blanco, necesita un nombre de archivo `.xml` para guardar el informe en este formato.
- Después de los resultados es necesario un parámetro `--report`. Tras un espacio en blanco, lleva un nombre de archivo `.html` para guardar un informe "legible por humanos" en este formato (este es el que podemos ver en *SCAP Workbench).*
- Finalmente, después de un espacio en blanco, necesita la ruta al archivo `.xml` con la política de seguridad a usar con el sistema operativo actual. Estos archivos están en el archivo de políticas

descargado de *GitHub* que descomprimimos, y es el que define el nombre del perfil de seguridad especificado en la opción `--profile` (en nuestro caso es el archivo `ssg-ubuntu1804-ds-1.2.xml`)

🔍 ***Asegúrate de que el informe que obtengas tiene los mismos resultados que la ejecución equivalente con SCAP Workbench (a menos que hayas corregido algo antes, en este caso el informe probablemente tendrá más controles de seguridad solucionados). Solo tienes que probar esto con uno de los tres perfiles que utilizaste en el ejercicio anterior.***

# Bloque 2. Remediación con *OpenSCAP*

## Ejercicio L6B2_OSCAPREMWB: Remediación con *SCAP Workbench*

### Descripción de la actividad / Aplicación práctica

Necesitas hacer ***hardening* automático a tu SO** siguiendo las políticas del paquete `scap-security-guide` usando remediación OSCAP

### Resultados Esperados

Esta actividad finalizará cuando utilices `oscap` o *SCAP Workbench* para intentar **corregir los controles de seguridad que no pasan sus comprobaciones**. Una vez que tengas los resultados, debes **comparar la puntuación *de Lynis* y el % de las reglas SCAP aplicadas correctamente** para determinar si la corrección tuvo un efecto positivo, y determinar a partir de tus resultados si crees que los perfiles de *Lynis* y SCAP miden cosas diferentes (en otras palabras, si usan diferentes controles de seguridad)

### Otra información necesaria para su realización

**Primero, ejecuta una auditoría con *Lynis* para tener una puntuación de seguridad base y almacena el resultado en un archivo que puedas consultar más adelante.** Vuelve a seleccionar el perfil CIS e intenta corregir automáticamente sus controles de seguridad marcando "*Remediate*" en el GUI. Compara y analiza los resultados con la ejecución inicial del perfil CIS de la sección anterior. **Ejecuta la auditoría *Lynis* de nuevo y almacena el informe en un archivo diferente.** Compara la puntuación *Lynis* y el porcentaje de reglas SCAP aplicadas correctamente antes y después de la corrección. *¿Han mejorado?*

## Ejercicio L6B2_OSCAPREMBASH: Remediación con *scripts* de *bash*

### Descripción de la actividad / Aplicación práctica

Necesitas hacer ***hardening* automático** a tu SO siguiendo las políticas del paquete `scap-security-guide` usando remediación **con *scripts* de *bash***

### Resultados Esperados

Esta actividad finalizará cuando **uses los *scripts* de corrección** `bash` incluidos en el fichero de directivas de seguridad descargado para intentar corregir los controles de seguridad que no pasen sus comprobaciones. También debes comparar de nuevo el % de reglas aplicadas correctamente y las puntuaciones de *Lynis* antes y después de la remediación, *¿crees ahora que los perfiles SCAP y Lynis miden cosas diferentes?*

### Otra información necesaria para su realización

Vete a los perfiles de seguridad descargados y descomprimidos en una carpeta al principio del laboratorio y accede a la carpeta `bash` dentro de ella. Localiza el *script* de shell correspondiente al perfil de seguridad CIS de *Ubuntu,* **hazlo ejecutable** y ejecútalo con privilegios de `root` para ver cuántos controles de seguridad

se han corregido ahora. Audita el SO con el perfil de seguridad de nuevo **sin corrección automática** (a través de *SCAP Workbench* o línea de comandos) y compara el nuevo informe con el anterior. Además, haz de nuevo una auditoría *Lynis* y compárala con la hiciste en el ejercicio anterior.

# Ejercicio L6B2_OSCAPREMANS: Remediación con *Ansible*

## Descripción de la actividad / Aplicación práctica

Necesitas hacer ***hardening* automático a tu SO** siguiendo las políticas del paquete `scap-security-guide` usando remediación **con *Ansible***

## Resultados Esperados

Esta actividad finalizará cuando **uses los *scripts* de corrección de *Ansible*** incluidos en el archivo de directivas de seguridad descargados para intentar corregir los controles de seguridad que no superen sus comprobaciones. También debes comparar nuevamente el % de reglas aplicadas correctamente y las puntuaciones de *Lynis* antes y después de la remediación para determinar qué procedimiento de remediación obtiene mejores resultados.

## Otra información necesaria para su realización

El procedimiento de remediación final incluido en los archivos `oscap` es a través del programa de automatización de configuraciones ***Ansible***. Para ejecutar las políticas de corrección con *Ansible* (ya instalado en la máquina virtual de la asignatura, pero instalable con: `sudo apt install ansible`), debes hacer el siguiente procedimiento:

- Vete a los perfiles de seguridad descargados, accede a la carpeta `ansible` y localiza el archivo `.yml` correspondiente al *perfil CIS*
- Edita este `.yml` y cambia `hosts: all` por `hosts: localhost`
- Ejecuta `sudo ansible-playbook ubuntu1804-playbook-cis.yml`
- Ejecuta una verificación de la política nuevamente con `oscap` y una auditoría con *Lynis* para ver si los resultados mejoran

# Bloque 3: Auditoría y remediación con STIGs

## Ejercicio L6B3_STIGSCCAUDIT: Auditoría automatizada con *SCAP Compliance Checker* (SCC) y STIGs

### Descripción de la actividad / Aplicación práctica

Consiste en usar el programa *SCAP Compliance Checker* (SCC) y los STIG que vienen con él para **auditar un sistema *Ubuntu* de acuerdo con las especificaciones de los STIG de la DISA** (*Defense Information Systems Agency*) de EEUU.

### Resultados Esperados

Puedes usar la herramienta mencionada para **auditar tu MV *Ubuntu* y generar un informe con sus resultados**, entendiendo el proceso y la información que la herramienta devuelve. Puedes contestar a estas preguntas, además de las que se plantean en el ejercicio:

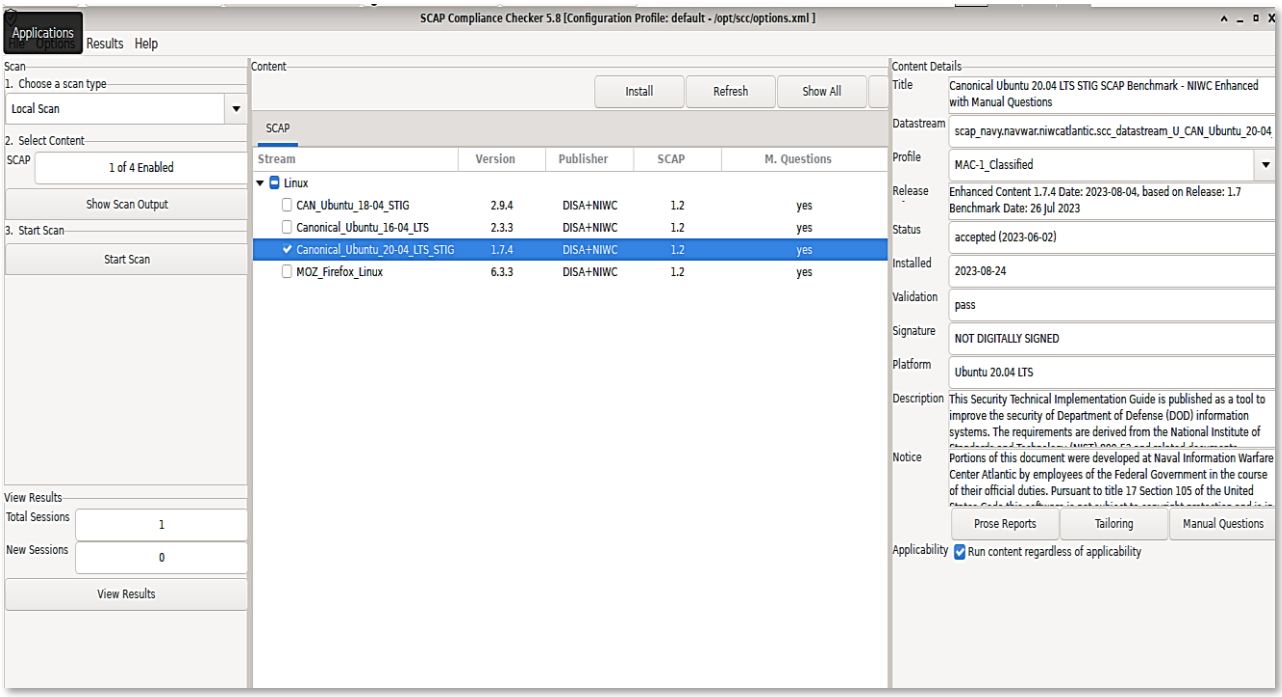
- *¿Localizas en la herramienta los niveles MAC y demás elementos definidos para los STIGs en la teoría, de forma que puedas elegir bajo qué condiciones pasas la auditoría?*
- *¿Te das cuenta de que el informe generado no solo te da una evaluación, sino que te detalla todo lo que tienes que hacer para mejorar los controles que el sistema no pasa?*
- *¿Entiendes que esta herramienta hace auditoría automática, pero no remediación?*

### Otra información necesaria para su realización

Si visitas https://public.cyber.mil/stigs/scap/ y buscas entre el contenido **SCAP 1.2**, verás que muy recientemente EEUU ha puesto a disposición general para algunos productos **una herramienta de automatización SCAP oficial que, hasta ahora, solo permitía usar a proveedores de su gobierno**. Esta herramienta la mencionamos en teoría y se llama **SCC** (*SCAP Compliance Checker*). Podemos descargar en nuestra *MV Ubuntu 22.04* la herramienta usando este enlace: https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/scc-5.8_ubuntu22_amd64_bundle.zip

Descarga por tanto la herramienta y descomprime el zip. En el verás un archivo `.deb` que tiene la instalación de esta herramienta. En un GUI *Gnome* bastaría con hacerle doble clic a ese fichero para empezar a instalarla, pero en XFCE4 tenemos que hacer vía consola con: `sudo dpkg -i <ruta del fichero .deb>`

Una vez instalada la herramienta, podrás ejecutarla con `sudo /opt/scc/scc` y verás su interfaz (ignora si hay alguna advertencia).




Como verás, por defecto la herramienta incluye 4 STIGs, tres de ellos correspondientes a versiones pasadas de ***Ubuntu*** y uno de ellos a ***Firefox***. Antes de usarlos, haz clic por favor el botón ***Refresh***.

> 🔍 ***Es posible que si la descargas ahora ya aparezca el STIG de Ubuntu 22.04. Si es así, úsalo en lugar del que te vamos a mencionar ahora.***

Para probar la herramienta, **marca solo el STIG de *Ubuntu* 20**, aunque no sea la versión correcta de tu SO, y asegúrate de que la opción "***Run content regardless of applicability***" está marcada para que SCC escanee el SO igualmente. Localiza el sitio donde se pueden configurar los perfiles de escaneo y mira que están todas las combinaciones vistas en la teoría.

> 🔍 ***Para esta prueba basta con seleccionar una MAC 1 Classified.***

Cuando lances el escaneo en el botón de la izquierda el sistema te advierte de que ese STIG tiene algunos controles **que te hacen preguntas que debes responder manualmente,** y como crear las respuestas. No lo vamos a hacer en esta prueba, por lo que podemos simplemente ***Continuar*** para escanear el sistema con todo lo que sea automatizable.

Espera a que termine la auditoría del sistema y se genere un informe. Para poder verlo desde la herramienta necesitas instalar otra herramienta aparte, pero no es necesario en nuestro caso. Simplemente busca en el directorio ***home* del usuario actual**, y abre el fichero HTML que encontrarás en `SCC/Sessions/<fecha y hora en la que se lanzó el informe>`. Dentro del informe, accede a "***All Settings***" y contesta las siguientes preguntas:

- *¿Hay una puntuación general de compliance del sistema?*
- *¿Un Ubuntu recién instalado está cerca de alcanzar ese compliance con los STIG de la DISA o más bien lejos?*
- *¿Están los resultados clasificados por su CAT claramente y de manera fácilmente accesible en forma de lista de problemas a resolver?*
- *¿Tiene cada resultado el detalle suficiente como para saber solucionar el problema que describen dentro del propio informe?*

🔍 ***DISA sí que publica contenidos para hacer remediación automática, pero no con las características del protocolo SCAP, sino con Ansible, Chef (una herramienta que sirve para lo mismo que Ansible, desplegar configuraciones de forma automatizada) o directamente Powershell. Puedes encontrar los que están disponibles públicamente aquí: https://public.cyber.mil/stigs/supplemental-automation-content/.***

***Desgraciadamente, no lo hay para todos los sistemas cubiertos por los STIGs, solo para unos pocos (por ejemplo, para Ubuntu 22 aún no hay disponible ninguno, al menos en Diciembre de 2023)***

# 🛠 Ejercicio L6B3_STIGSCCREM: Ejemplo de remediación con *SCAP Compliance Checker* (SCC) y STIGs para *Firefox*

## 🥷 Descripción de la actividad / Aplicación práctica

Consiste en **aprender más sobre las opciones de SCC** con el perfil que tiene para *Firefox*, incluyendo **actualizar la herramienta con nuevos STIGs** y aprender a pasarla de nuevo cuando se remedie algún problema para ver los resultados.

## 🎖 Resultados Esperados

Puedes **evaluar *Firefox* con SCC y aprender a mejorar su seguridad** siguiendo las indicaciones de su STIG DISA para mejorar su seguridad general. Puedes contestar a estas preguntas:

- *¿Entiendes por qué tanto énfasis en "blindar" un navegador?*
- *¿Comprendes que se puede mejorar la seguridad de Firefox con las opciones que pongas en un único fichero de políticas?*
- *¿Crees que esto facilita el despliegue automático de Firefox seguros?*

## 📖 Otra información necesaria para su realización

SCC es un programa que **admite nuevos STIGs** que vayan saliendo con el tiempo o que no vengan de serie con la herramienta (como podría ser uno de *Ubuntu 22* que la DISA publicase en el futuro). Para poder saber los que hay disponibles, hay que ir de nuevo a https://public.cyber.mil/stigs/scap/ y buscar entre el contenido SCAP 1.2. Hazlo, y **encuentra la última versión del STIG de *Firefox* para *Linux* disponible**. Descárgate su zip a tu máquina virtual.

Una vez tengas ese STIG, vete a SCC y usa el botón "***Install***" que tienes encima de la lista de STIGs actual. Ahora solo tienes que buscar el zip que te has descargado y seleccionarlo para que la herramienta lo incorpore a su lista. Fíjate que en este caso es una versión posterior al que trae de serie.

Asegúrate de que **solo está marcado el STIG de *Firefox* nuevo** y, antes de lanzarlo como en el ejercicio anterior, **selecciona como perfil a evaluar *MAC 1 Classified*** y que lo ejecute de todas formas aunque no sea aplicable. *¿Qué resultado da?*

Ya que el resultado del escaneo es manifiestamente mejorable, vamos a aprender **cómo securizar *Firefox*** en este entorno para que, si quieres, puedas hacer una política de seguridad que puedas luego aplicar cuando quieras. Resulta que cuando arrancas *Firefox* el programa está preparado para buscar un fichero llamado `policies.json` en una ruta predeterminada. En caso de *Ubuntu* es `/etc/firefox/policies`. Si

creas esa carpeta, puedes crear dentro de ella ese fichero siguiendo estas instrucciones: https://support.mozilla.org/en-US/kb/customizing-firefox-using-policiesjson.

Ahora, mira el informe que ha generado SCC e **implementa algunas de las soluciones sugeridas por el STIG** que involucren cambiar ese fichero, añadiendo una directiva y un valor siguiendo dicho informe. Una vez que hayas incorporado un par de ellas, **vuelve a evaluar *Firefox*** y contesta a estas preguntas:

- *¿Está SCC mirando la configuración de tu Firefox o directamente mira el contenido de este fichero de políticas y nada más?*
- *¿Qué harías si consiguieses un fichero* `policies.json` *que devolviese una puntuación muy alta y tuvieras que crear una MV para clonar luego que permitiese la navegación segura?*

# Bloque 4: Auditoría y remediación con *CIS Benchmarks*

## Ejercicio L6B4_CISUBUNTUPRO: La licencia *Ubuntu Pro*

### Descripción de la actividad / Aplicación práctica

Consiste en sacarle partido a las funcionalidades de *Ubuntu Pro* a la hora **de hacer auditoría y remediación automatizada de controles de seguridad CIS** a un sistema *Ubuntu*.

### Resultados Esperados

Puedes usar las funcionalidades de *Ubuntu Pro* para auditar y remediar los controles de seguridad recomendados por el CIS para un sistema Ubuntu, **usando un perfil adecuado para el sistema auditado**. Puedes contestar a las preguntas que se hacen en el ejercicio y además a estas:

- *¿Te parece que la licencia Ubuntu Pro hace muy sencillo el hardening de un sistema Ubuntu?*
- *¿Pierde el sistema Ubuntu su GUI al hacer hardening?*
- *Tras examinar la documentación, ¿sabrías hacer un proceso equivalente para hacerle hardening a Ubuntu usando los controles de seguridad STIG de la DISA (Defense Information Systems Agency)? ¿Qué crees que pasaría ahora si le pasáramos el SCC del bloque anterior una vez hecho el proceso de fix correspondiente? (no se trata de hacerlo, sino de saber cómo se haría y sus consecuencias)*
- *Además, ¿sabrías como generar un fix para un sistema Ubuntu que no tiene instalados los paquetes necesarios para hacer este ejercicio con las herramientas proporcionadas? (no se trata de hacerlo, sino de saber cómo se haría)*

### Otra información necesaria para su realización

Para hacer este ejercicio **es necesario que te hagas una cuenta personal en *Ubuntu Pro*** (es gratuita): https://ubuntu.com/pro . Una vez la hagas y confirmes tu email (puedes dar el de Uniovi) puedes entrar en tu cuenta y en "***Your Subscriptions***" localizar tu "***Free Personal Token***". Este es el token que necesitas para seguir.

🔍 ***Ubuntu Pro es gratuito hasta para 5 máquinas. Si te preocupa "quemar" uno de tus 5 usos en esta prueba, no te preocupes. Se trata de 5 máquinas activas. Si una máquina no contacta con el servidor de Ubuntu Pro en unas 24 horas, se le dará automáticamente de baja en tu lista de máquinas activas, por lo que no perderás uno de tus usos*** 😊

Para hacer este ejercicio sigue estos pasos:

- Por si acaso, **no te olvides de hacer un *snapshot*** o un clon enlazado de la MV para poder volver a un estado base si algo va mal.
- **Haz una auditoría al sistema base con** `lynis` y guarda el informe generado, igual que en el **Laboratorio 1**.

- **Instala el cliente de *Ubuntu Advantage*** (UA) y adjunta el token de tu suscripción como indica aquí: https://ubuntu.com/security/certifications/docs/disa-stig/installation
- **Ejecuta un proceso de auditoría completo CIS** (sin remediar los controles de seguridad) siguiendo estas instrucciones: https://ubuntu.com/security/certifications/docs/usg/cis/audit. Usa el **perfil adecuado,** teniendo en cuenta que queremos **controles que no afecten a la funcionalidad del *Ubuntu*** y que la VM es una ***Server***: https://ubuntu.com/security/certifications/docs/usg/cis/compliance. Contesta a estas preguntas:
  - *¿Una MV Ubuntu base está más cerca de tener un % de compliance aceptable con las políticas CIS y con las STIG de la DISA del bloque anterior?*
  - En función de eso, *¿Cuál de las dos te parece más estricta en cuanto al nivel de seguridad imponen?*
- Espera a que termine la auditoría y **analiza el informe SCAP** generado en `/var/lib/usg/`, que tiene el mismo formato que el que vimos con *OpenSCAP* en el bloque 1. Ten en cuenta que el informe se genera con permisos de `root` y en un carpeta protegida, por lo que tendrás que cambiar el fichero de propietario para poder leerlo sin problemas. *¿Está el sistema lejos de tener un Compliance aceptable con los controles CIS?*
- **Ejecuta el procedimiento de `fix`** siguiendo estas instrucciones: https://ubuntu.com/security/certifications/docs/usg/cis/compliance. Contesta ahora a estas preguntas:
  - *¿Tienes conectividad a Internet?* Si la respuesta es no, *¿Entiendes por qué los procedimientos de remediación automática deben ejecutarse con precaución?*

🔍 ***En este caso, uno de los fixes hace que se desconfigure el DNS, pero se puede arreglar así: https://www.ionos.com/digitalguide/server/configuration/change-dns-server-on-ubuntu/ . Puedes poner el de Google, el NextDNS del laboratorio 1, o cualquiera*** 😊

  - Muchos de los *scripts* de terceros que aplican los controles del CIS **desinstalan el GUI** de un *Ubuntu Server*, bien por error o bien porque solo implementan los perfiles más estrictos. *¿Te ha pasado con las funcionalidades de Ubuntu Pro? ¿Por qué crees que ha sido?*
- **Pasar de nuevo un proceso de auditoría** y ver de nuevo el informe generado. *¿Está el sistema lejos de tener un Compliance aceptable con los controles CIS?*

## 🛠 Ejercicio L6B4_CISWAZUH: *Wazuh* como gestor de políticas CIS

### 🕵 Descripción de la actividad / Aplicación práctica

Consiste en usar la instalación de *Wazuh* que hemos hecho en el laboratorio anterior para poner en marcha la **evaluación automatizada de los sistema monitorizados con políticas CIS** vinculados a ellas.

### 🏅 Resultados Esperados

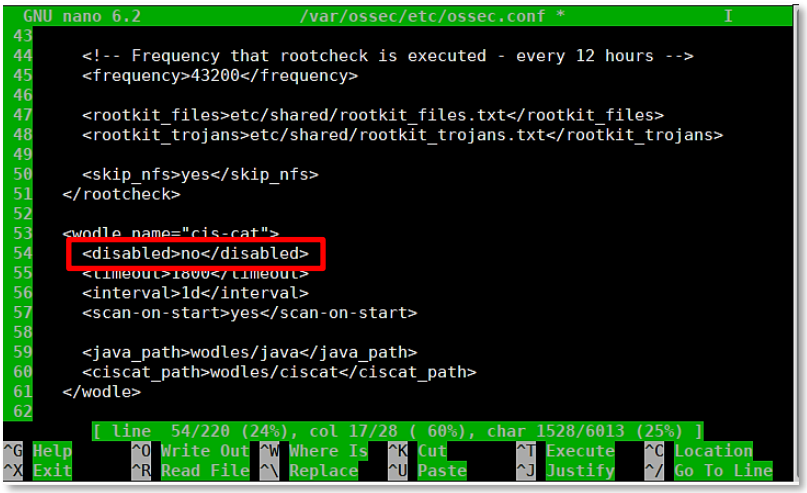
Puedes hacer que ***Wazuh* te evalúe el sistema que monitoriza** con las política CIS asociada al mismo y con ello **obtener un índice de seguridad CIS**, qué controles de seguridad pasa y cuáles no. Puedes contestar a estas preguntas:

- *¿Considera Wazuh que si el sistema monitorizado tiene una puntuación muy baja es un evento de seguridad del que hay que informar?*

- *¿Puedes obtener un listado de los controles de seguridad que el sistema no supera y todos los detalles de cada control (auditoria, remediación, descripción...), equivalentes a lo que tiene el PDF del CIS Benchmark correspondiente?*

## Otra información necesaria para su realización

Una de las grandes ventajas de usar *Wazuh* es que viene con todo un **sistema automatizado de gestión de auditorías con *CIS Benchmarks* integrado**. No obstante, para que podamos usarlo de la forma más efectiva, conviene que **modifiquemos el fichero de configuración del OSSEC** del agente en el sistema monitorizado para activar la integración con la herramienta de evaluación **CIS-CAT** que mencionamos en la teoría. No te olvides reiniciar el agente después de hacer el cambio.




Si todo es correcto, ahora deberías poder hacer clic en el agente dentro de *Wazuh* y usar la opción SCA (***Security Configuration Assessment***). Busca dentro de ella el informe realizado del sistema, su puntuación y si puedes encontrar todos los detalles de los controles que el sistema monitorizado no supera.

# ✍️ Insignias y Autoevaluación

**NOTA:** Tienes una versión de esta tabla de insignias en formato editable disponible en el *Campus Virtual*. Puedes usar este archivo para crear un documento en el formato que desees y tomar notas extendidas de tus actividades para crear un log de lo que has hecho. Recuerda que el material que elabores se puede llevar a los exámenes de laboratorio.

| Nivel de Insignia | Desbloqueada cuando | ¿Desbloqueada? |
|---|---|---|
|  | Puedes utilizar los perfiles de seguridad disponibles *de OpenSCAP* para cualquier producto soportado | |
|  | Puedes cargar un perfil de seguridad adecuado para un sistema operativo soportado | |
|  | Puedes utilizar *OpenSCAP* para intentar solucionar automáticamente posibles problemas en la aplicación de los controles de seguridad. Puedes responder a esta pregunta. *Incluso si no mejora significativamente el índice de seguridad de una máquina recién instalada, ¿qué crees que sucedería con una máquina que no se ha mantenido o administrado correctamente?* | |
|  | Puedes utilizar los archivos de perfiles de seguridad *de OpenSCAP* | |
|  | Puedes responder a las siguientes preguntas: *¿Qué sucede si un control de seguridad aparece en un informe como "no aplicable"? ¿Cuáles son las posibles razones? ¿Sigue siendo útil este control de seguridad para mejorar la seguridad del sistema?* | |
|  | Puedes realizar un análisis automatizado de un sistema operativo para detectar vulnerabilidades de acuerdo con un determinado perfil de seguridad. | |
|  | Puedes utilizar `oscap` desde la línea de comandos para analizar y corregir los controles de seguridad | |
|  | Puedes responder a la siguiente pregunta: *¿Los STIG de la DISA incluyen perfiles correspondientes a todas las combinaciones posibles de MAC y niveles de confidencialidad que vimos en teoría?* | |
|  | Puedes responder a las siguientes preguntas: *¿SCC sirve para auditar y remediar o solo para auditar? ¿Cómo podrías mejorar la seguridad en caso de que sea solo auditoría?* | |
|  | Puedes utilizar los scripts de corrección `bash` de cualquier directiva de seguridad SCAP. | |
|  | Puedes utilizar los scripts de corrección de *Ansible* de cualquier directiva de seguridad SCAP. | |
|  | Puedes crear un procedimiento de corrección para *Firefox* siguiendo la política de seguridad de los STIGs de la DISA | |
|  | Puedes analizar y comprender el contenido de un informe `oscap` | |
|  | Comprendes las ventajas y desventajas de las herramientas de *Compliance as Code*, y cómo intentar alcanzar de esta forma altas puntuaciones de seguridad pueden tener efectos secundarios que puede | |

| | | |
|---|---|---|
| | impedir que las máquinas funcionen correctamente y requieran intervención manual | |
|  | Puedes responder a esta pregunta: *¿qué método automatizado tiene mejores resultados de corrección en el perfil CIS de Ubuntu en OpenSCAP: corrección de Oscap, corrección de script bash o corrección de Ansible? ¿La mejora de la seguridad es significativa a partir de un sistema recién instalado?* | |
|  | Entiendes cómo aprovecharte de las ventajas de seguridad que te da la licencia de *Ubuntu Pro* y como automatizar la remediación de problemas de seguridad | |
|  | Comprendes como *Wazuh* se puede usar para saber con mucha precisión el nivel de seguridad de cada una de las máquinas que tiene auditadas siguiendo los perfiles CIS | |
|  | **This is the way**: Tu dominio de los procedimientos de *hardening* automatizado te permite mejorar la seguridad de un sistema *Ubuntu* de acuerdo con varias políticas de seguridad que garantizan diferentes niveles de esta. Además, puedes evaluar automáticamente la seguridad de cualquier sistema *Ubuntu* u otro *software* con herramientas profesionales de referencia mundial. Con esto, estás preparado para usar los mismos procedimientos en cualquier otro sistema que admita la misma clase de automatización. | |