Examen Final – OpenScap y Nessus

Yeinel De Los Santos

2024-1204

Seguridad Informática

Instituto Tecnológico de Las Américas

Seguridad de Sistemas Operativos

Profesor Juan Alexander Ramírez

Agosto 14, 2025

Enlace al video: https://youtu.be/yGomWZtxCEU

Lista de reproducción:

https://www.youtube.com/playlist?list=PLRDZZD2E5jNRY-AAvkBXurV25Ul7nVbY0

Introducción

El dia de hoy estaremos viendo nuestro examen final, el objetivo de este examen es ver el uso de la herramienta OpenScap, la cual es un conjunto de utilidades de código abierto cuyas siglas dicen: Security Content Automation Protocol, es un estándar que nos permite evaluar nuestro sistema en busca de vulnerabilidades y nos permite aplicar diferentes parches de seguridad basados en varios perfiles de seguridad como el que estaremos usando, siendo este el perfil: PCI-DSS

Estaremos combinando esta herramienta con otra que vimos en el parcial pasado, Nessus, con estas herramientas escanearemos y resolveremos automáticamente los problemas que puedan ser detectados dentro de nuestra maquina cliente y de esa manera completaremos este examen.

Para la realización de esta evaluación estaremos cumpliendo las siguientes pautas:

En esta práctica deben implementar OpenSCAP y automatizar los parches de seguridad requeridos por PCI-DSS, el cual es el estándar de seguridad de datos para la industria de tarjetas de crédito y entidades procesadoras de pago.

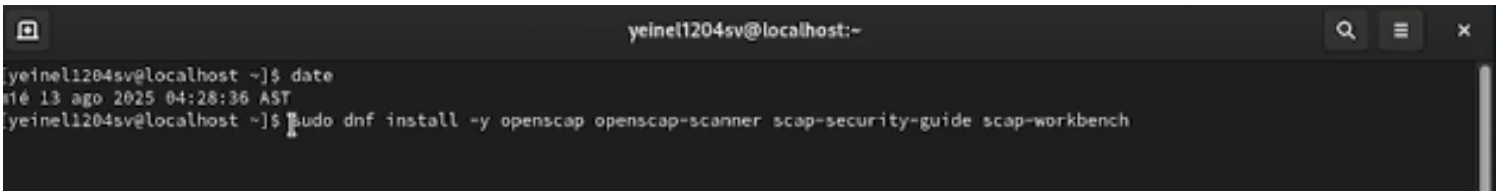
- Instalar OpenSCAP en un servidor Linux de su preferencia (utilizar nomenclatura (Nombre + 4 ultimos digitos matricula)
- Instalar otro equipo con Linux como cliente (utilizar nomenclatura (Nombre + 4 ultimos digitos matricula)
- Realizar un escaneo de vulnerabilidades al equipo cliente utilizando Nessus, sino hay vulnerabilidades deben provocarlas.
- Inicializar el SCAP workbench en el servidor y elegir el perfil de PCI-DSS
- Escanear el equipo cliente y Aplicar los parches de seguridad requeridos por el estándar PCI-DSS
- Escanear nuevamente con Nessus y OpenScap y exportar el reporte de vulnerabilidades.

Nota: Si existen vulnerabilidades no corregidas previamente por OpenSCAP, debe detallarlas y ofrecer una solución

Nota: Deben validar que la evaluacion se haga de manera correcta, ya que sino se conecta bien presentara un N/A en los perfiles y eso es como que no se haya hecho o encontrado nada. Sin mas que explicar, comencemos:

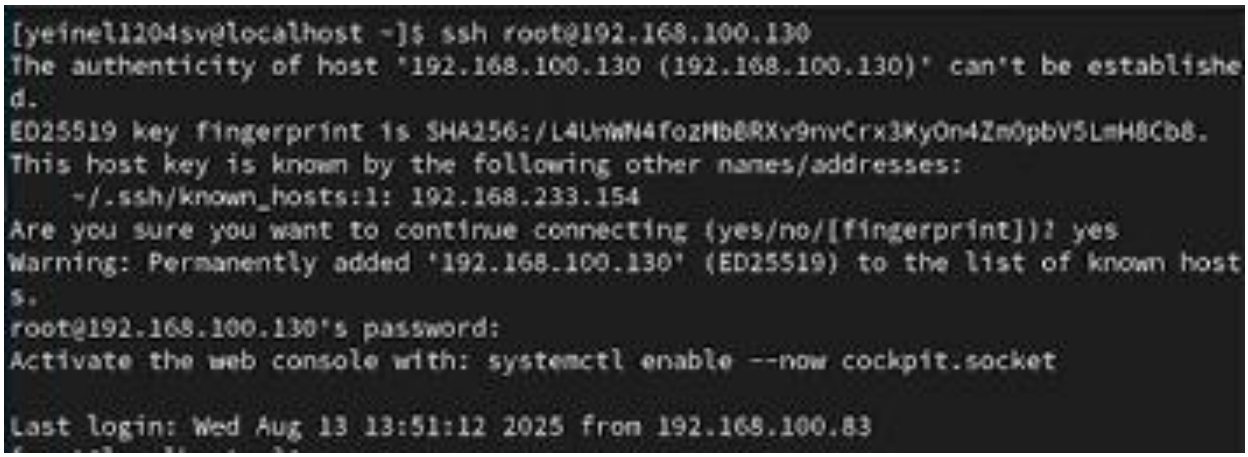
Instalación, preparación y escaneos

Empezando con esta práctica lo que debemos tener primordialmente es Nessus y OpenScap listos para poder empezar con esto, como anteriormente estuvimos mostrando la instalación de Nessus en el parcial anterior, veremos ahora mismo como podemos instalar OpenScap, el cual es tan simple como usar el comando sudo dnf install -y openscap openscap-scanner scap-security-guide scap-workbench.



Así es como tendríamos instalado el openscap sin ninguna complicación mayor.

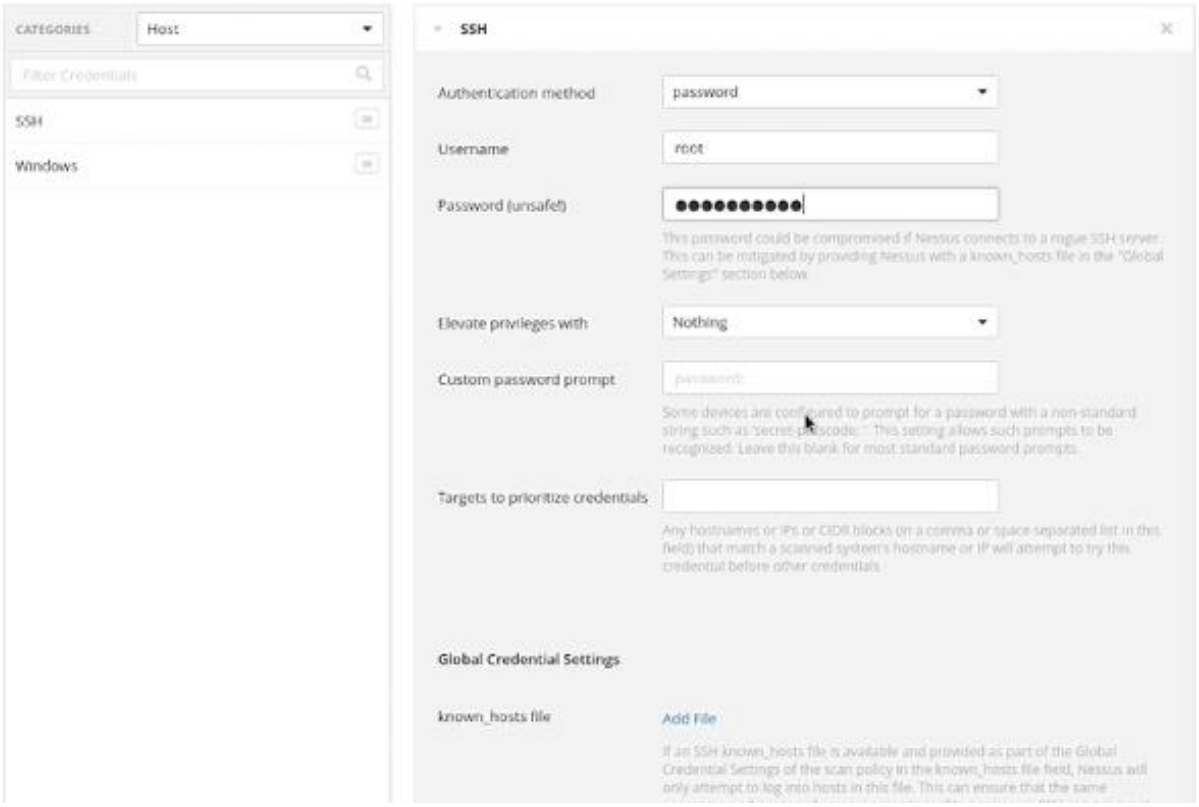
Luego de asegurarnos de tener todo instalado y con el ssh funcionando, verificándolo con el comando ssh root@192.168.100.130



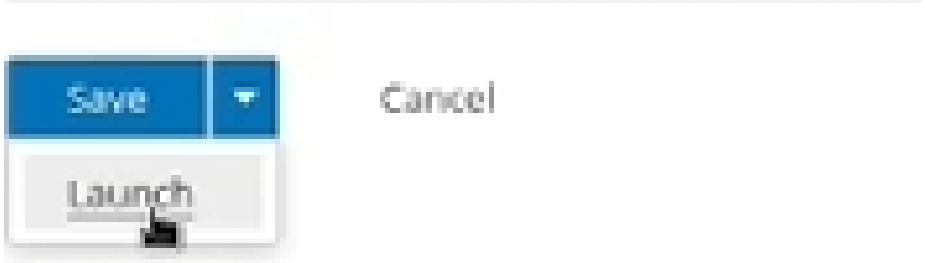
Ya podemos empezar con el examen en si, lo primero que haremos será realizar dos escaneos a la maquina cliente para ver el estado en el que podemos encontrar esta desde el inicio, tal vez con alguna vulnerabilidad pequeña o algo extra que pongamos en caso de que lo necesitemos.

Dentro de Nessus primero haremos un escaneo de vulnerabilidades como sabemos hacerlo, ponemos el nombre del escaneo, en este caso, este será el ScanOne, ponemos la dirección IP y esta vez como no estamos haciendo el escaneo en Windows haremos algo un poquito diferente.

Lo que haremos será darle a la opción donde nos dice que accederemos por ssh, donde al hacerlo, veremos cómo vamos a otra sección donde nos pedirá las credenciales que queramos usar, en este caso yo usare las credenciales con contraseña, donde pondremos el usuario root y su contraseña.



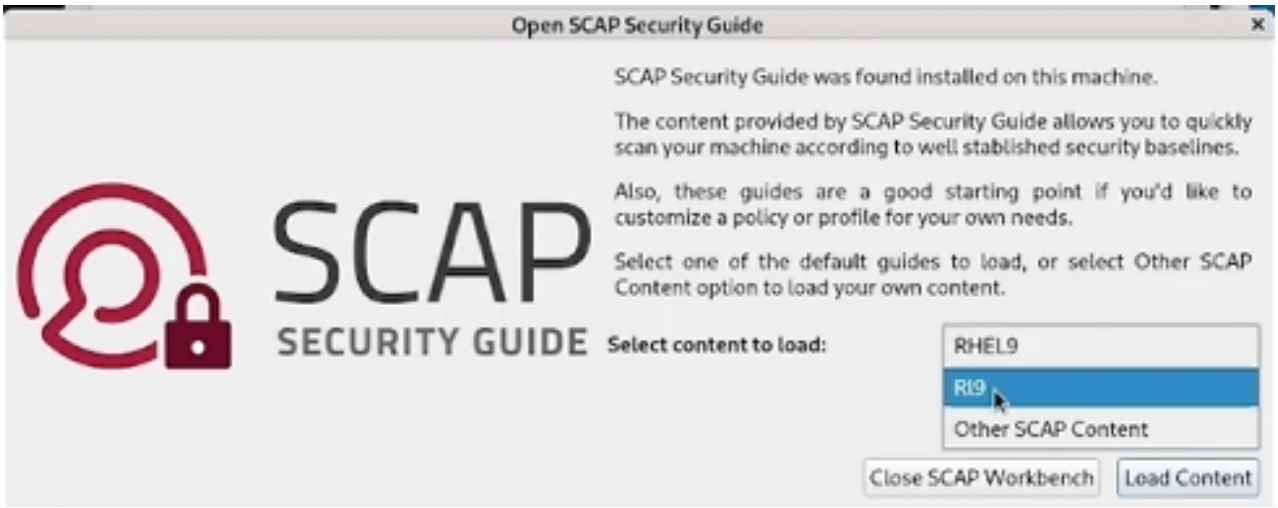
Despues de confirmar esos 3 datos, simplemente vamos a la parte baja y le damos a lanzar el escaneo.



Luego de haber lanzado el primer escaneo con Nessus, ahora debemos lanzar nuestro primer escaneo con OpenScap.

Lo primero que haremos será abrir el programa, se puede hacer tanto desde la terminal con el comando scap-workbench, luego de que este sea lanzado, nos veremos frente a frente con su menú, donde como estamos en Rocky Linux por temas de compatibilidad, ya que los diseñadores principales de este programa son personas del equipo de red hat o basados en red hat, esta herramienta se encuentra principalmente en estas distribuciones.

Al entrar en el menú nos dará diferentes opciones con las cuales debemos cargar el contenido, como estamos en la versión de rocky linux 9.6, le daremos a la versión RL9 y le daremos a cargar contenido.



Luego de que carguemos el contenido, se nos abrirá el siguiente menú dentro del programa.

Menú el cual nos da muchas opciones de como hacer las cosas dentro del programa, tenemos una opción para guardar personalizaciones, tenemos los perfiles de seguridad que podemos elegir, tenemos el objetivo tanto de maquina local como maquina remota por ssh, el puerto por el que podemos acceder, también podemos marcar si el usuario es un sudoer y podemos ver las reglas aplicadas por los perfiles de seguridad elegidos por nosotros mismos.

Aquí podemos ver el menú

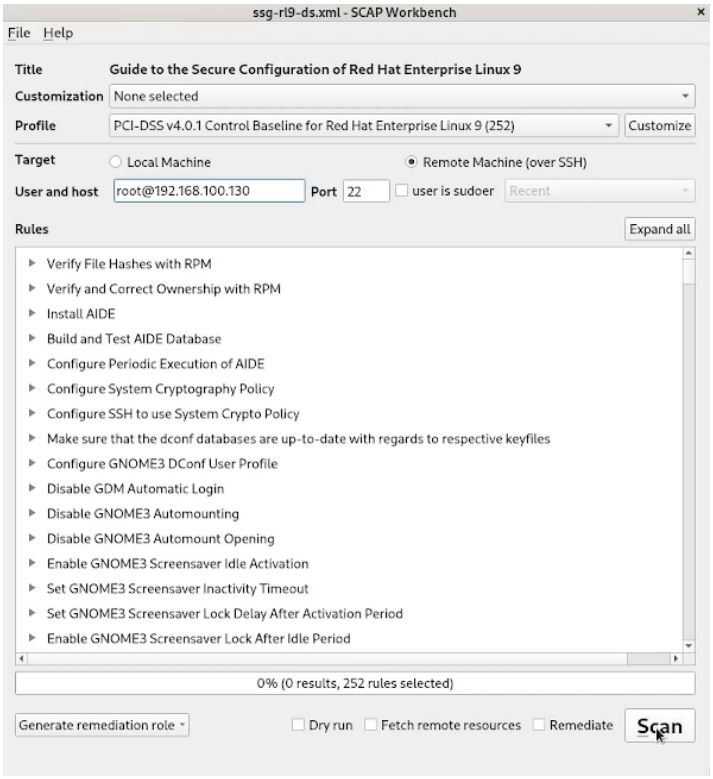
En este menú aplicaremos diferentes configuraciones para hacer el escaneo y posteriormente, OpenScap pueda solucionar los problemas detectados por el mismo.

Las configuraciones que colocaremos serán las siguientes:

- En perfil usaremos el estándar PCI-DSS el cual es un estándar de seguridad diseñado para proteger los datos de las tarjetas de crédito y débito durante las transacciones y el almacenamiento
- El objetivo que le pondremos al escáner será una maquina remota por ssh y cuando nos active la opción pondremos las siguientes credenciales: root@192.168.100.130,.

Dejaremos todo lo que queda de la manera predeterminada y podremos ahora sí, empezar el escaneo.

Despues de la configuración correcta, aquí podemos ver las opciones colocadas en el escáner.

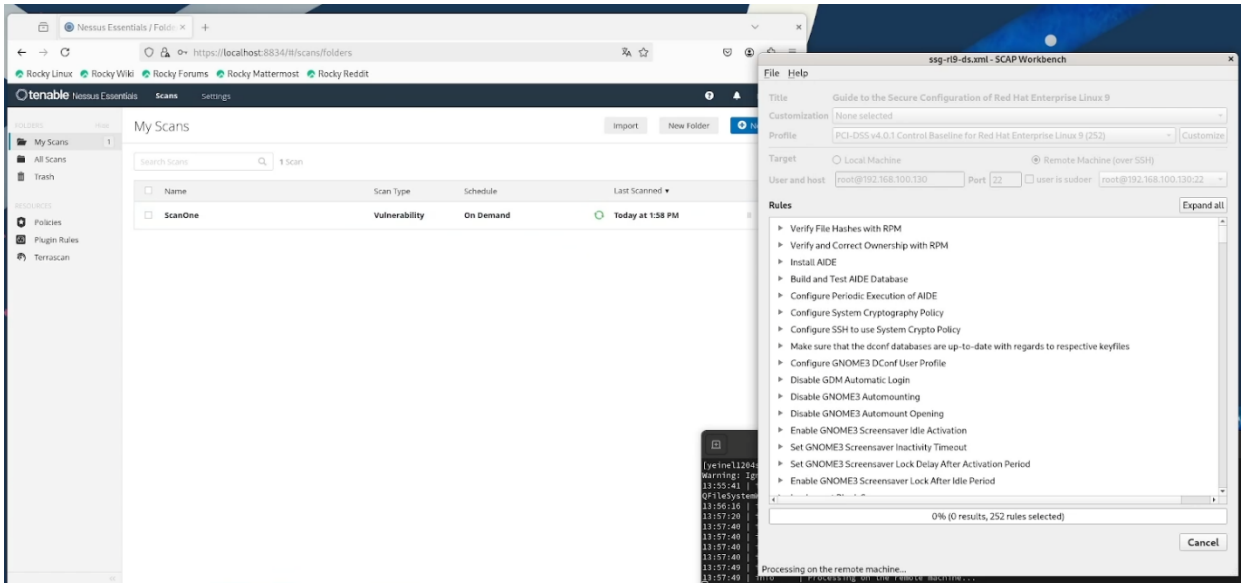


Antes de empezar el escaneo, veremos como OpenScap nos pide la contraseña del usuario root y la pondremos correctamente para poder comenzar.

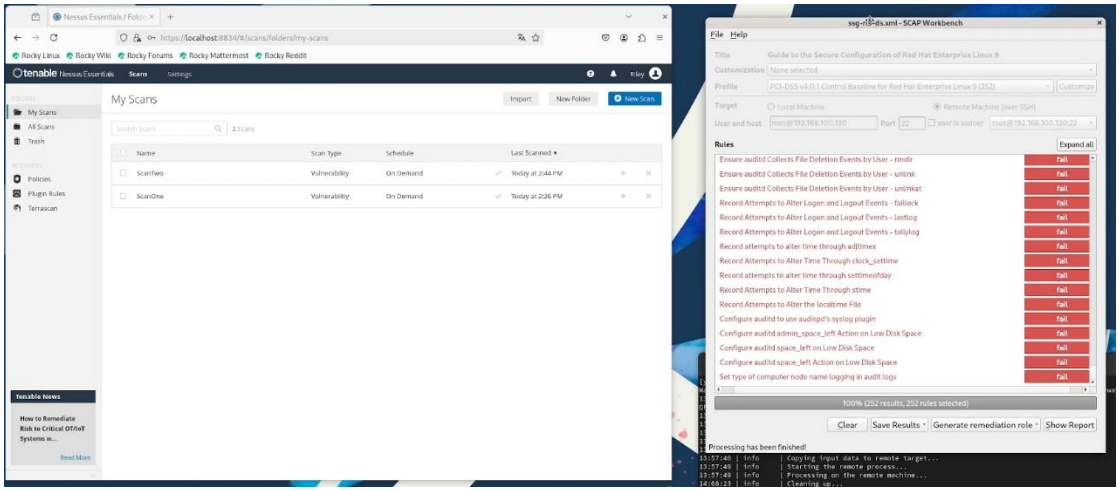
Iniciaremos el escaneo y esperaremos a ambos escáneres terminen sus asignaciones.

Despues de que los escáneres hayan terminado con sus tareas cada uno, veremos el primero escaneo de todos para detectar el estado principal de la maquina antes de las alteraciones que vamos a hacer para vulnerar mas el equipo.

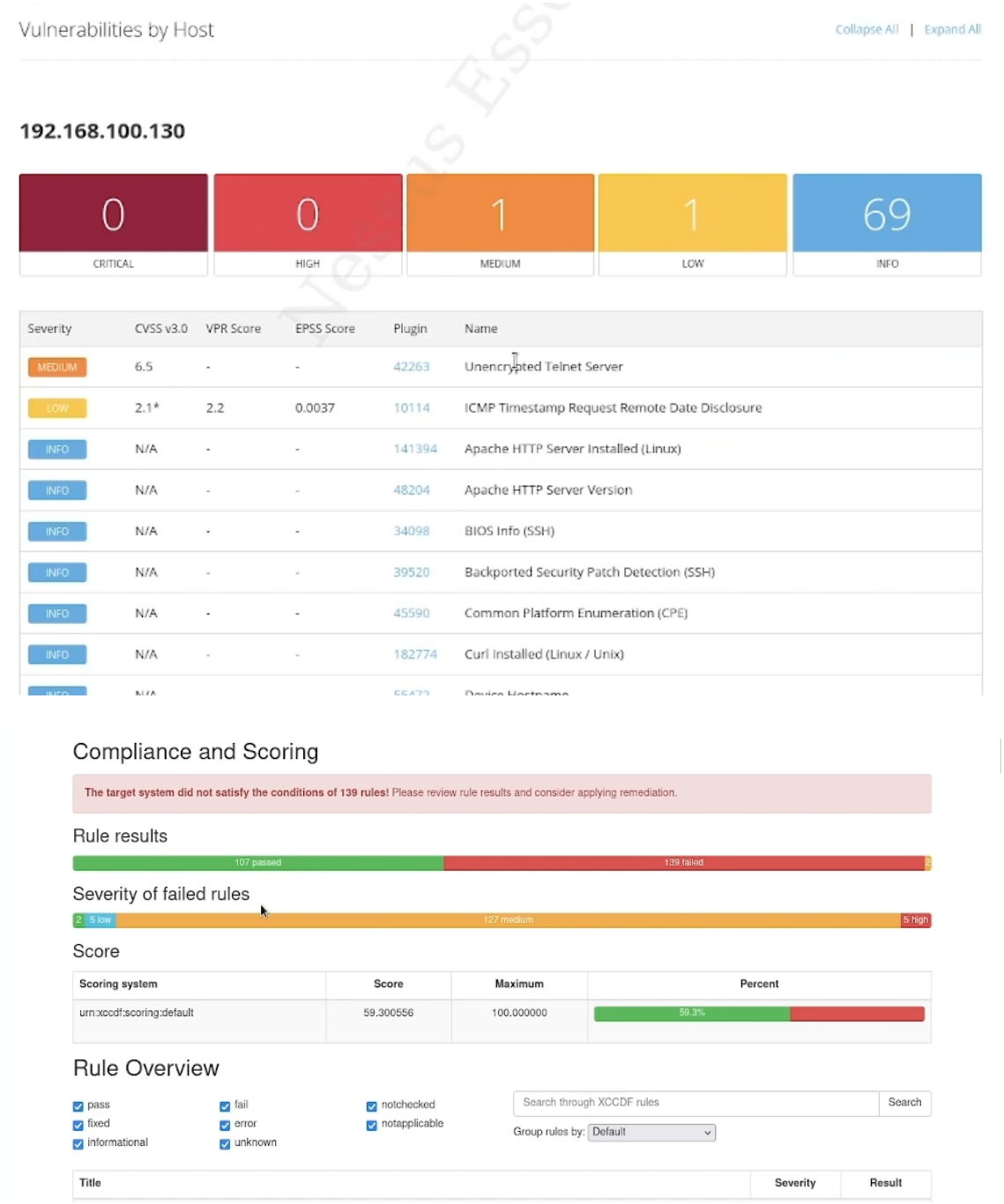
Escaneos en proceso



Escaneos completados



Primer escaneo; OpenScap y Nessus



Con los primeros escaneos completados podemos ver que Nessus no detecta tantas fallas como lo podría detectar OpenScap, este ultimo al ser hecho especialmente para estos sistemas, detecta vulnerabilidades mas profundas y es capaz de arreglarla por si mismo, ahora que hemos visto lo que es el primer escaneo en sí, añadamos algunas vulnerabilidades para ver cómo pueden incrementar y como puede OpenScap ayudar con todo esto.

Configuración de vulnerabilidades

Para proseguir con el examen, necesitamos configurar algunas opciones dentro de la maquina para que nuestros escáneres detecten más vulnerabilidades y poder medir mejor las cosas, así como de paso completamos una de las pautas de la practica como lo puede ser crear mas vulnerabilidades dentro del sistema, veremos una pequeña descripción, así como una foto con los comandos que usaremos para instalarlos en nuestro sistema.

1. Servicio de telnet

El servicio telnet es un protocolo se considera obsoleto por su falta de seguridad, especialmente la ausencia de cifrado de datos, lo que lo hace vulnerable a ataques como el sniffing y el acceso no autorizado.

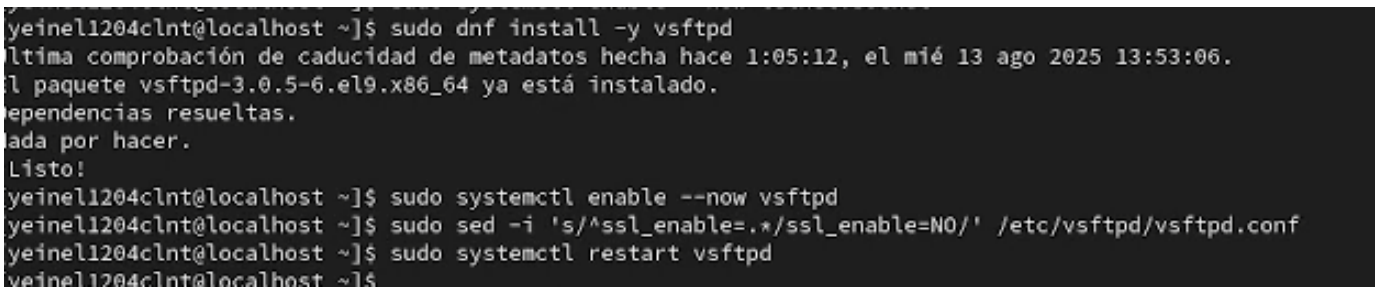
Comando de instalación:

2. Vsftpd

Vsftpd (Very Secure FTP Daemon), es un servidor FTP (Protocolo de Transferencia de Archivos) diseñado para sistemas tipo Unix y Linux.

Un servidor Vsftpd mal configurado puede causar muchas fallas dentro del sistema, entre ellos podeos ver ataques de denegación de servicios, así como ataques de puerta trasera.

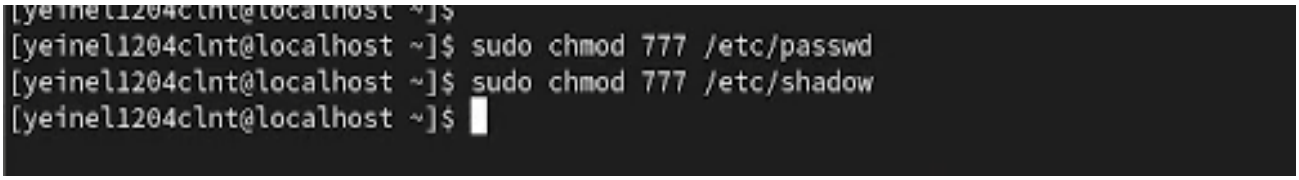
Comandos para instalar y dejar SSL activado:



3. Carpetas sensibles

Tenemos carpetas sensibles como passwd que contiene todas las informaciones sensibles de las cuentas de usuario dentro del sistema o la carpeta shadow donde se encuentran todas las contraseñas cifradas e información sobre el vencimiento de estas.

Comando para darle permisos de escritura, lectura y ejecución a todos los usuarios.

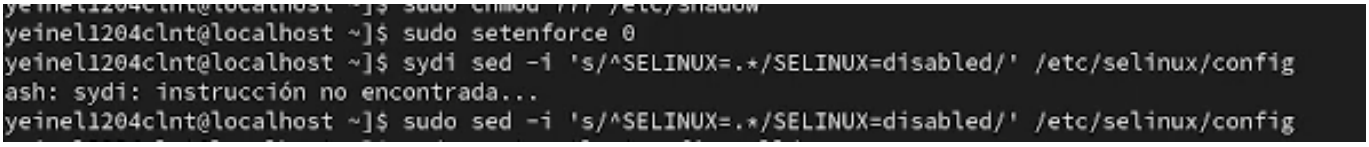


4. SELinux

SELinux es un módulo de seguridad para el núcleo de Linux que implementa un sistema de control de acceso obligatorio.

Si desactivamos esta configuración tendremos un problema mayor de seguridad donde incluso se pueden ver involucrados los daemons del sistema

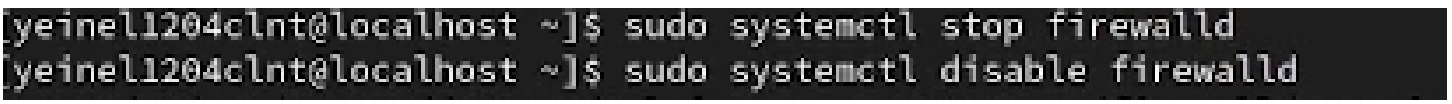
Comandos para desactivar SELinux:



5. Firewall

Como todos sabemos, el firewall es uno de los sistemas de protección mas importantes dentro de lo que es cualquier sistema, si terminamos desactivando esta opción, es igual a tener el sistema al aire libre frente a cualquier amenaza

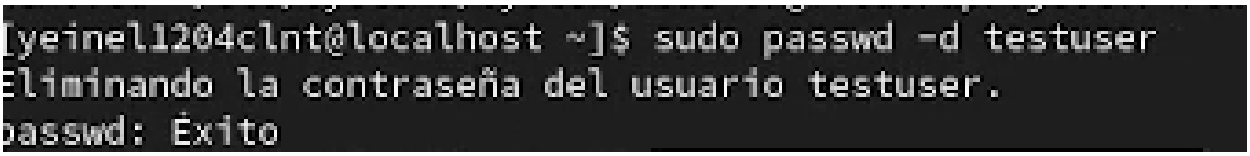
Comandos para desactivar y detener el firewall:



6. Usuarios sin contraseñas

Los usuarios sin contraseñas son riesgos que no podemos permitirnos, dependiendo del nivel de permisos que tengamos en ese usuario, se convierte en una amenaza aun mayor, en esta parte tenemos un usuario llamado testuser y le quitamos la contraseña en este usuario.

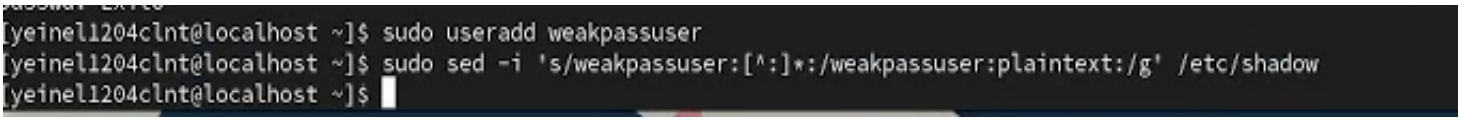
Comando para quitar la contraseña:



7. Usuarios con contraseñas sin cifrar

Otro de los problemas que pueden ocasionar las malas configuraciones de los usuarios, podemos incluir las contraseñas en texto plano, donde por razones de seguridad como las de tener las contraseñas al aire libre, atribuyen a las razones por las cuales no confiarnos con esto.

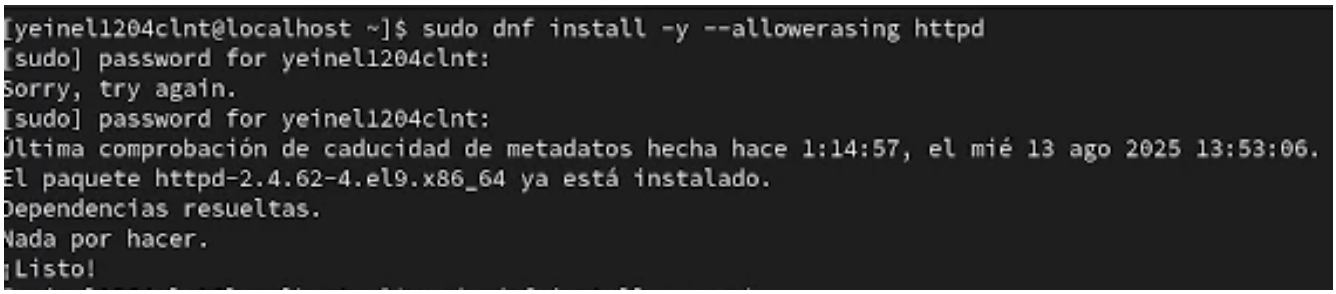
Comando para crear un usuario y agregarle una contraseña en texto plano:



8. Servicio httpd

Este servicio en contraparte con su versión mejorada (https) es un servicio que, si es activado sin ningún tipo de configuración, puede hacer que nuestro sistema sea vulnerado y que pueda acceder a través de los puertos de este hacia el interior de nuestra máquina que lo aloje

Comando para instalar el servicio:



9. Servicio Samba

Samba es un software de código abierto que permite a las computadoras con sistemas operativos Linux/Unix compartir archivos e impresoras con equipos Windows y viceversa.

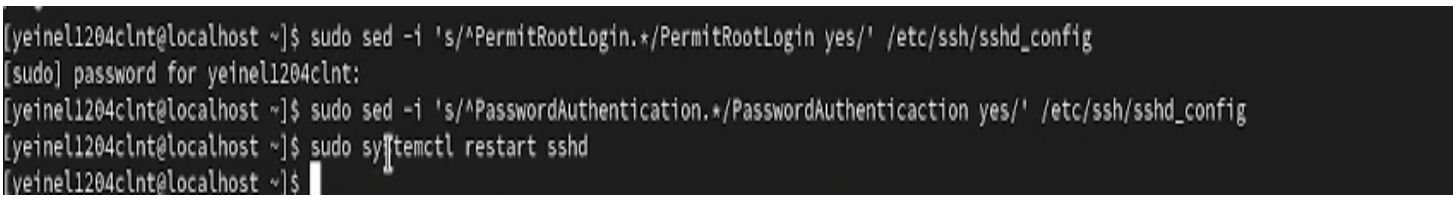
Este servicio mal configurado y activado puede ocasionar muchas vulnerabilidades que dejan diferentes formas de entrar al sistema a los atacantes, funcionando como una puerta trasera para inyectar código e interceptar la comunicación entre equipos

Comandos para instalar y desconfigurar samba:

10. SSH mal configurado

El servicio ssh para las conexiones remotas es uno de lo mejores servicios que existen en su tipo, sin embargo, como muchos servicios como el, podemos ver que, si lo configuramos mal, estamos abriendo nuestras puertas para que, si alguien llega a atacarnos, puede llegar a escalar de privilegios dentro de un sistema hasta llegar incluso a los privilegios root si no tenemos cuidado

Comandos para dejar que accedan al root de ssh y permitir la autenticación:

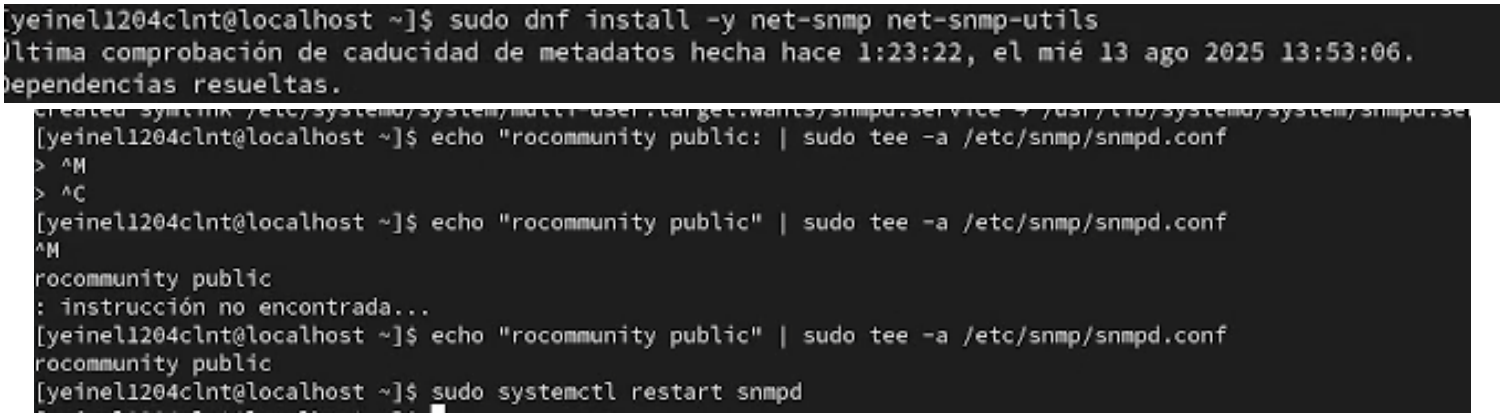


11. Servicio snmp

El protocolo SNMP es vulnerable principalmente debido a sus primeras versiones, SNMPv1 y SNMPv2, que transmiten datos sin cifrar y utilizan cadenas de comunidad en texto plano para la autenticación. Esto permite a los atacantes interceptar la información y, potencialmente, manipularla o tomar el control de los dispositivos gestionados.

Siendo esto algo tan conocido que reconocido que se han reportado muchas instancias de ataque DDoS hacia estos servicios, convirtiéndolos en un riesgo que no nos podemos permitir.

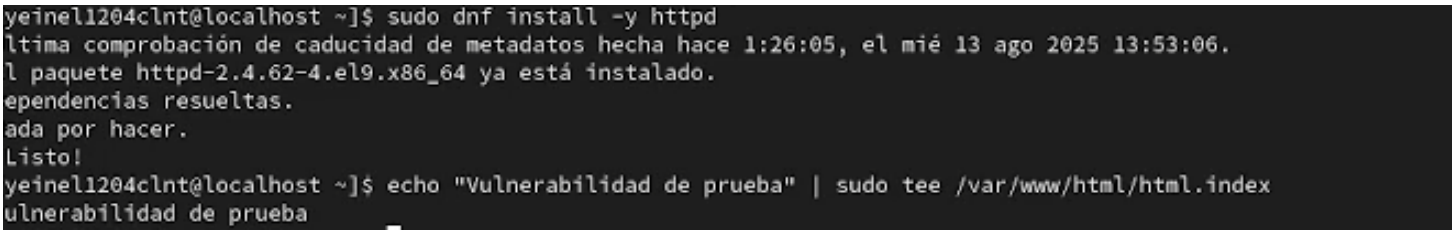
Comando para instalar y habilitar snmp para que acepte todo tipo de conexiones:



12. Paginas sin configurar en httpd

Aparte tener el servicio activo, podemos tener una pagina que no tenga ningún tipo de seguridad o controles, a través de esta pagina pueden pasar un sin numero de cosas que pueden afectar a nuestro sistema, en este caso dejaremos una pagina llamada vulnerabilidad de prueba.

Comando para crear la página simple:



13. Configuraciones extra en archivos de configuración e instalaciones extra.

Finalmente, dentro del aparta de vulnerabilidades que hemos esta explicando, hice un extra a la hora de hacer mas vulnerable el sistema, accediendo a los archivos de configuración y activando opciones las cuales permiten que sea una maquina aún más vulnerable, especialmente en httpd, snmp, sshd y Vsftpd.

También instale y active algunos servicios extra para dejar mas puertos abiertos dentro del sistema

Comandos para entrar a los archivos de configuración e instalación de otros servicios.

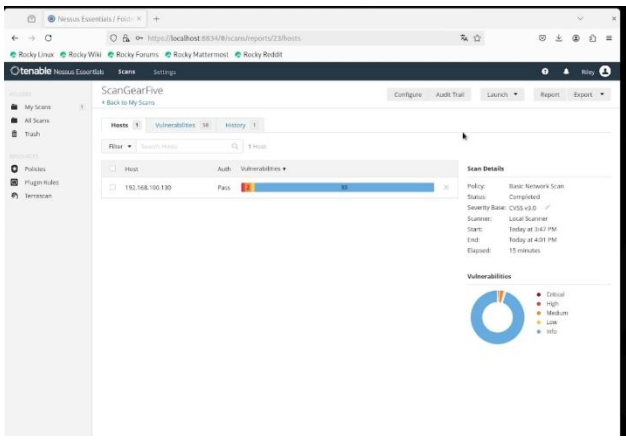
```
[yeinel1204clnt@localhost ~]$ sudo dnf install --allowerasing httpd-2.4.51
[sudo] password for yeinel1204clnt:
Ultima comprobación de caducidad de metadatos hecha hace 1:47:51, el mié 13 ago 2025 13:53:06.
No hay coincidencias para el argumento: httpd-2.4.51
Error: No se pudo encontrar ningún resultado: httpd-2.4.51
[yeinel1204clnt@localhost ~]$ sudo nano /etc/vsftpd/vsftpd.conf
[yeinel1204clnt@localhost ~]$ sudo nano /etc/ssh/sshd_config
[yeinel1204clnt@localhost ~]$ sudo useradd test
[yeinel1204clnt@localhost ~]$ echo "test:123456" | sudo chpasswd
[yeinel1204clnt@localhost ~]$ nano /etc/httpd/conf/httpd.conf
[yeinel1204clnt@localhost ~]$ sudo nano /etc/httpd/conf/httpd.conf
[yeinel1204clnt@localhost ~]$ sudo dnf install php php-mysqlnd mariadb-server httpd git
sudo systemctl enable mariadb --now
sudo systemctl enable httpd --now
git clone https://github.com/digininja/DVWA.git /var/www/html/dvwa
Ultima comprobación de caducidad de metadatos hecha hace 1:51:44, el mié 13 ago 2025 13:53:06.
El paquete httpd-2.4.62-4.el9.x86_64 ya está instalado.
Dependencias resueltas.
```

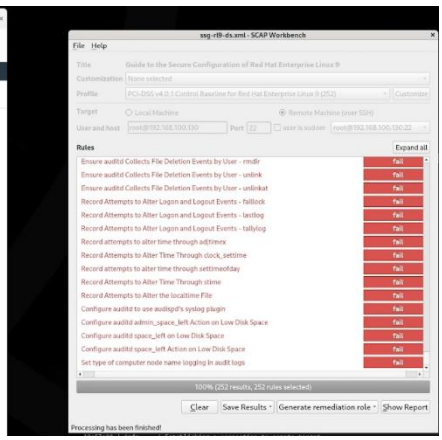
Bien, ya que hemos configurado los servicios tanto nativos como instalados del sistema de manera insegura, es hora de hacer unos nuevos escaneos para compararlos y ver que otras vulnerabilidades han podido encontrar los programas.

Siguiendo la misma línea de los primeros escaneos, vamos a comenzar nuevamente una ronda de escaneos para ver cómo está actualmente nuestra maquina cliente.

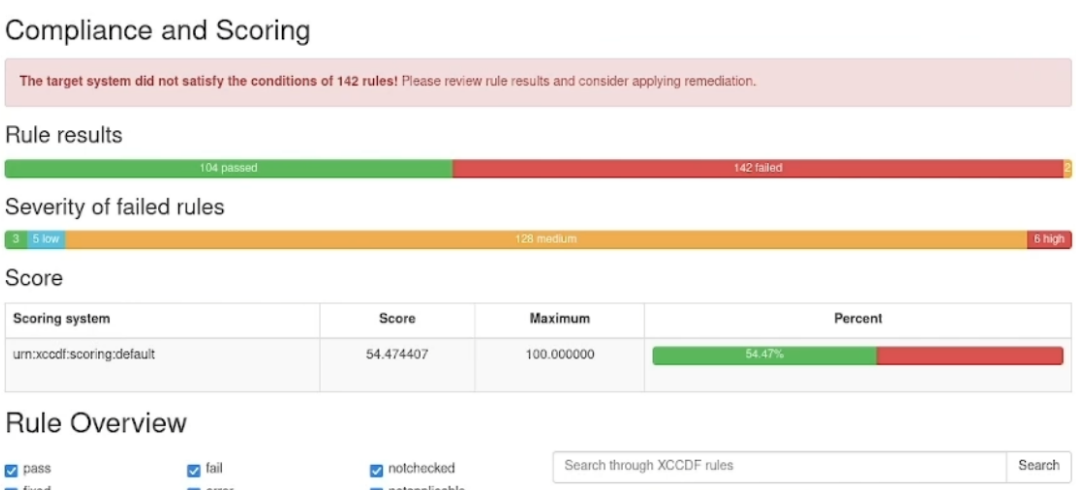
Segundo escaneo; OpenScap y Nessus

A la hora de verlos por encima, nos llama la atención como en Nessus tenemos directamente una vulnerabilidad alta y en OpenScap tenemos incluso menos secciones en pass.



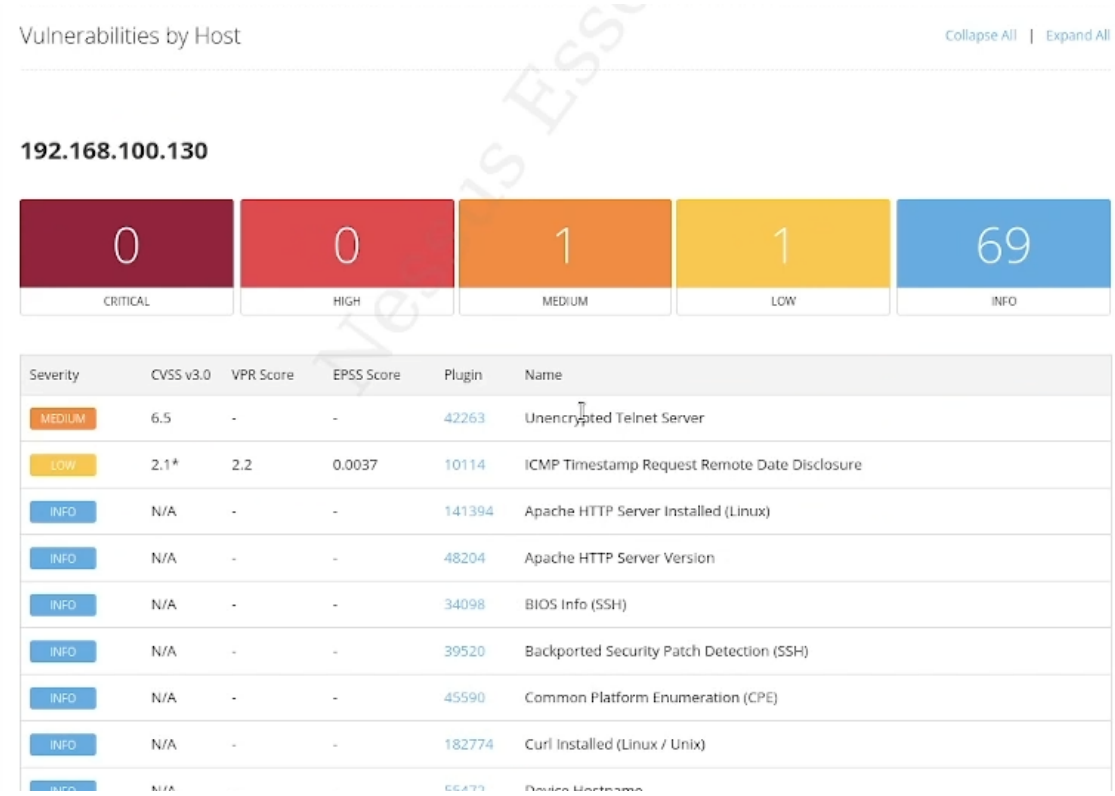


Despues vemos como nuestros escaneos en números tienen diferencias, por ejemplo, en Nessus vemos como detecto más vulnerabilidades, así como en OpenScap.



Comparaciones de cerca

Nessus:



Vulnerabilities by Host

Collapse All | Expand All

192.168.100.130

0	0	1	1	69
CRITICAL	HIGH	MEDIUM	LOW	INFO

Severity	CVSS v3.0	VPR Score	EPSS Score	Plugin	Name
MEDIUM	6.5	-	-	42263	Unencrypted Telnet Server
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	141394	Apache HTTP Server Installed (Linux)
INFO	N/A	-	-	48204	Apache HTTP Server Version
INFO	N/A	-	-	34098	BIOS Info (SSH)
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	182774	Curl Installed (Linux / Unix)

ScanGearFive

Wed, 13 Aug 2025 16:01:56 AST

TABLE OF CONTENTS

Vulnerabilities by Host

- 192.168.100.130

Vulnerabilities by Host

Collapse All | Expand All

192.168.100.130

0	1	2	1	76
CRITICAL	HIGH	MEDIUM	LOW	INFO

Show

OpenScap:

Compliance and Scoring

The target system did not satisfy the conditions of 139 rules! Please review rule results and consider applying remediation.

Rule results

107 passed | 139 failed | 2

Severity of failed rules

2 | 5 low | 127 medium | 5 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	59.300556	100.000000	59.3%

Rule Overview

pass | fail | notchecked
fixed | error | notapplicable
informational | unknown

Search through XCCDF rules | Search

Group rules by: Default

Title	Severity	Result

Compliance and Scoring

The target system did not satisfy the conditions of 142 rules! Please review rule results and consider applying remediation.

Rule results

104 passed | 142 failed | 2

Severity of failed rules

3 | 5 low | 128 medium | 6 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	54.474407	100.000000	54.47%

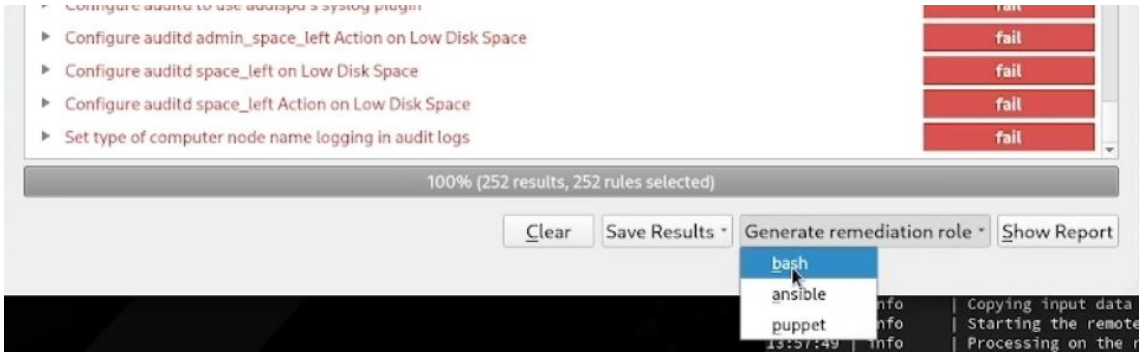
Rule Overview

pass | fail | notchecked

Search through XCCDF rules | Search

Ahora, ¿viendo que nosotros tenemos ya las vulnerabilidades y necesitamos arreglar nuestra máquina, como lo hacemos desde nuestra maquina cliente con OpenScap?

Sencillo, simplemente necesitamos generar un bash con OpenScap y pasarlo a la maquina cliente, lo que haremos será darle al botón de remediation role y elegimos la opción de bash.



Lo guardaremos en el directorio de nuestra elección e iremos a nuestra terminal.

Aquí iremos a el directorio o carpeta donde guardamos nuestro código y usaremos el comando scp para copiarlo a través de ssh y usaremos el usuario root para recibir el archivo en la maquina cliente a donde nosotros queramos, donde yo lo pasare a escritorio con el siguiente comando.

```
[yeinel1204sv@localhost ~]$ ls
Descargas  Documentos  Escritorio  Imágenes  Música  Plantillas  Público  Videos
[yeinel1204sv@localhost ~]$ cd Descargas
[yeinel1204sv@localhost Descargas]$ ls
Nessus-10.9.2-el9.x86_64.rpm        remediation.sh           ScanOne_ck4jss.html
Nessus-10.9.2-ubuntu1604_amd64.deb  ScanGearFive_kpblx2.html  ScanTwo_60xzt6.html
[yeinel1204sv@localhost Descargas]$ scp remediation.sh root@192.168.100.130:/home/yeinel1204clnt/Escritorio
root@192.168.100.130's password:
remediation.sh                                                100%  789KB  26.2MB/s   00:00
[yeinel1204sv@localhost Descargas]$
```

Luego de que hayamos copiado el archivo a nuestra maquina cliente, podremos ejecutarlo dentro de nuestro sistema simplemente usando el comando bash.

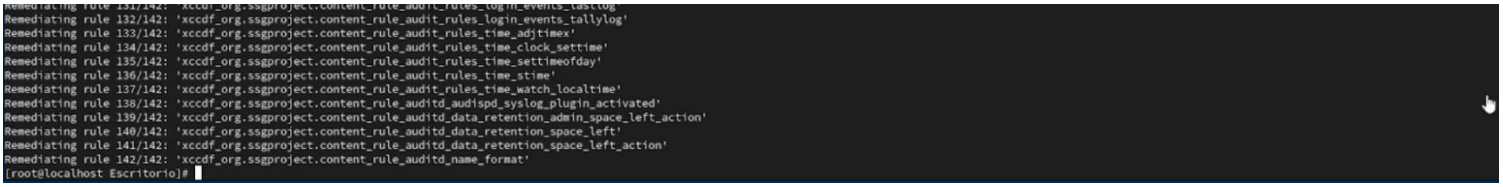
```
[yeinel1204clnt@localhost ~]$ ls
Descargas  Documentos  Escritorio  Imágenes  Música  Plantillas  Público  Videos
[yeinel1204clnt@localhost ~]$ cd Escritorio
[yeinel1204clnt@localhost Escritorio]$ ls
remediation.sh
[yeinel1204clnt@localhost Escritorio]$ su
Contraseña:
[root@localhost Escritorio]# ls
remediation.sh
[root@localhost Escritorio]# bash remediation.sh
Remediating rule 1/142: 'xccdf_org.ssgproject.content_rule_package_aide_installed'
Última comprobación de caducidad de metadatos hecha hace 2:17:15, el mié 13 ago 2025 13:53:06.
Dependencias resueltas.
================================================================================
 Paquete                Arquitectura            Versión                 Repositorio                Tam.
================================================================================
Instalando:
 aide                   x86_64                  0.16-103.el9            appstream                  146 k

Resumen de la transacción
================================================================================
Instalar  1 Paquete

Tamaño total de la descarga: 146 k
Tamaño instalado: 353 k
Descargando paquetes:
aide-0.16-103.el9.x86_64.rpm                    145 kB/s | 146 kB     00:01
--------------------------------------------------------------------------------
Total                                           106 kB/s | 146 kB     00:01
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
  Preparando          :                                                 1/1
```

Luego de haber ejecutado el comando, veremos como dependiendo del numero de errores que hayamos tenido en el escaneo de OpenScap, por ejemplo, el mío está parchando 142 errores que vimos en el segundo escaneo.

Podemos ver también directamente como va corrigiendo errores de poco a poco, lo que haremos será esperar a que termine las correcciones y vamos a ver con un escaneo nuevo todo lo que ha hecho OpenScap.



Ya en este punto después de esta imagen hemos visto como ha terminado el escaneo.

Lo que vamos a hacer después de esto es una tercera ronda de escaneos de Nessus y OpenScap para detectar y ver todos los cambios que ha hecho el programa dentro de la maquina cliente.

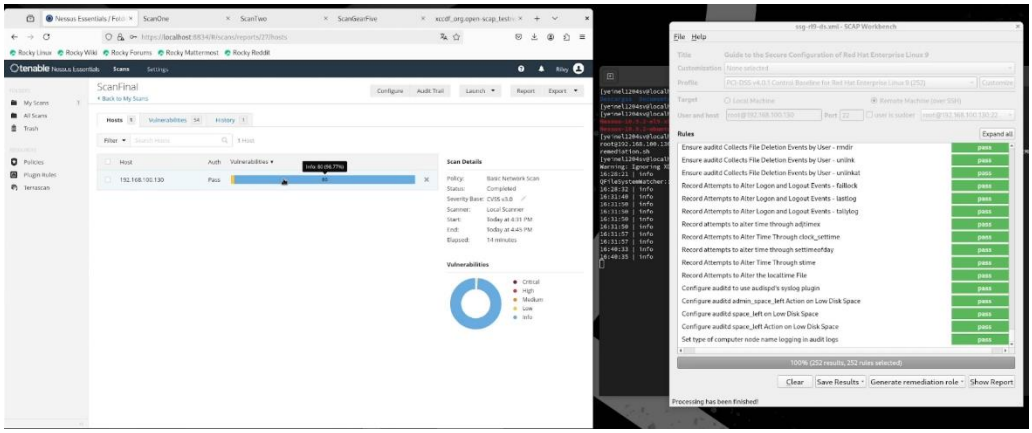
Nota: Revisar las contraseñas de root, ya que dependiendo de si es muy débil o si el sistema detecta que esta ha expirado, será necesario cambiarla para poder continuar accediendo al root por ssh.

Luego de haber terminado la última ronda de escaneos veremos como los cambios son masivos, ya que la mayoría de las vulnerabilidades fueron corregidas por OpenScap, incluso vulnerabilidades menores detectadas por Nessus.

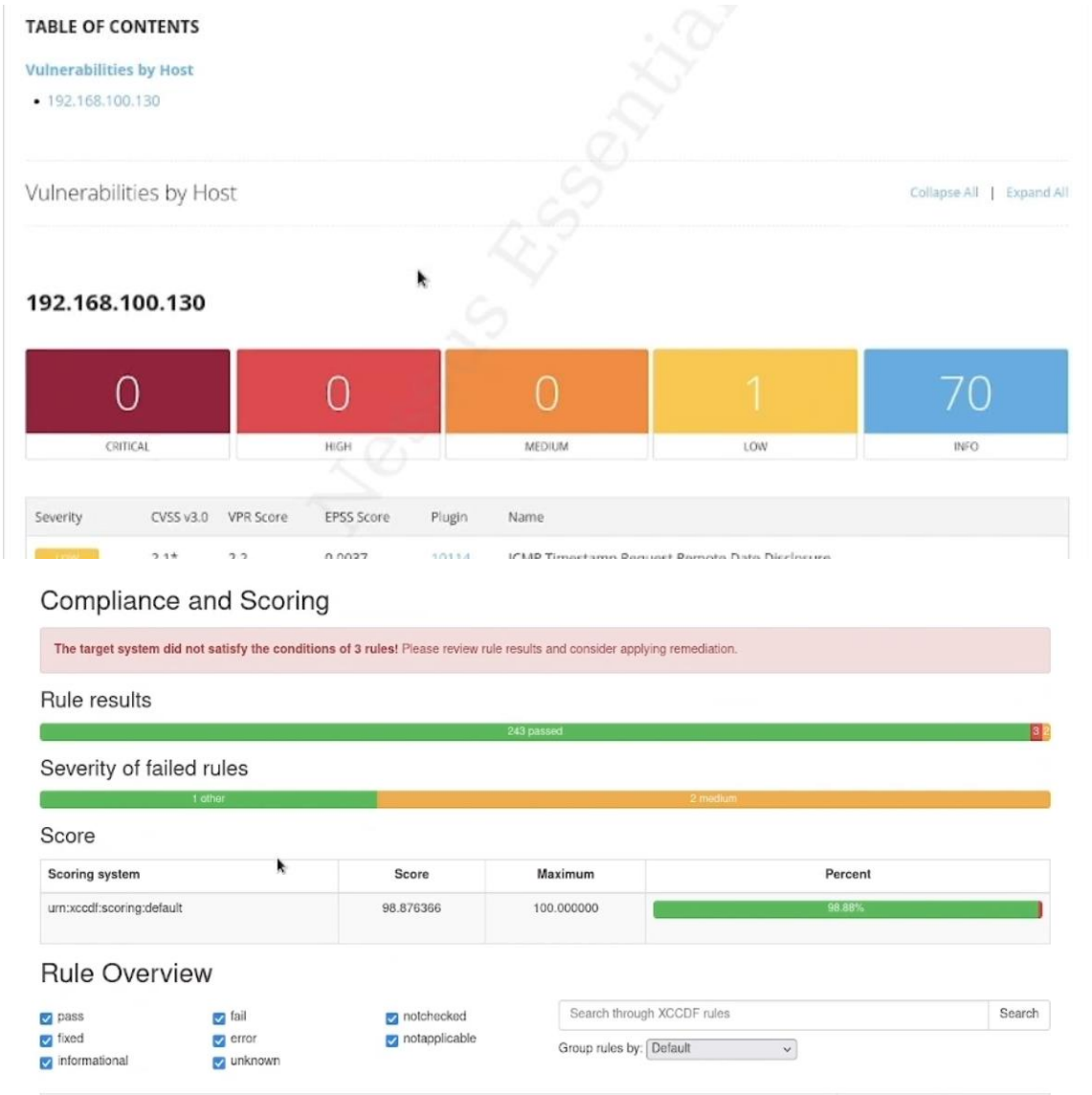
Las únicas vulnerabilidades que no pudieron ser corregidas por OpenScap son las que tienen un bloqueo, ya que las configuraciones para resolver estas vulnerabilidades, necesitamos que sea hecho por el usuario, para evitar tantos errores del sistema o también puede ser por datos que puedan interferir con el funcionamiento común del sistema si no son cambios hechos por el usuario.

Tercer y ultimo escaneo; OpenScap y Nessus

Aquí podemos ver como a primera vista tenemos un cambio drástico, Nessus ha disminuido sus detecciones de vulnerabilidades y vemos como los controles de seguridad de OpenScap están casi todos pasados.



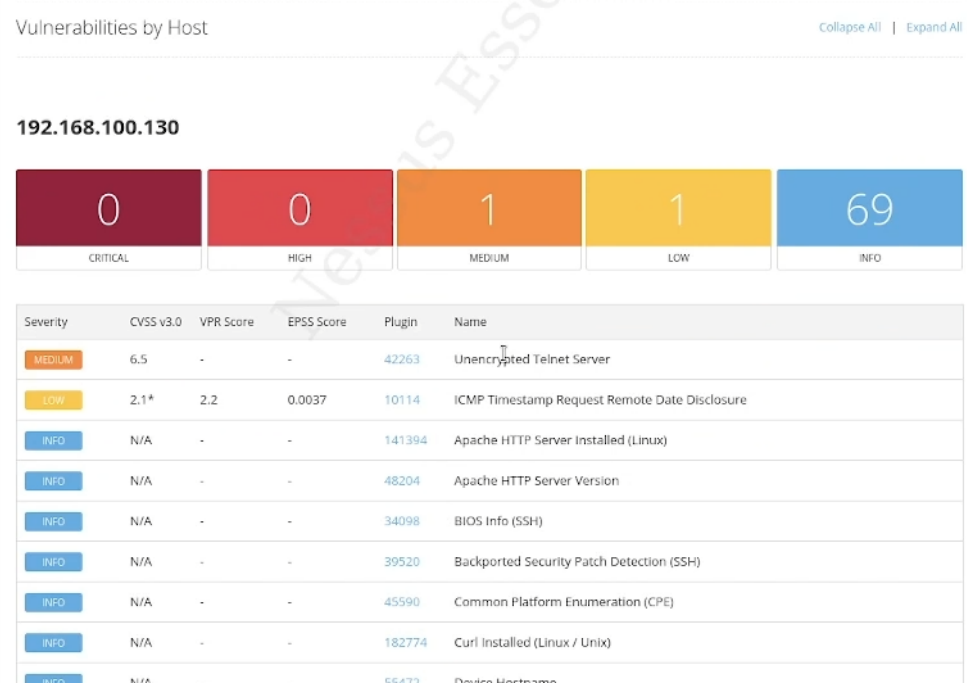
La mayor diferencia de todas las veremos en los reportes completos generados por ambos programas, donde notaremos una mejoría gigante.



Comparaciones Finales

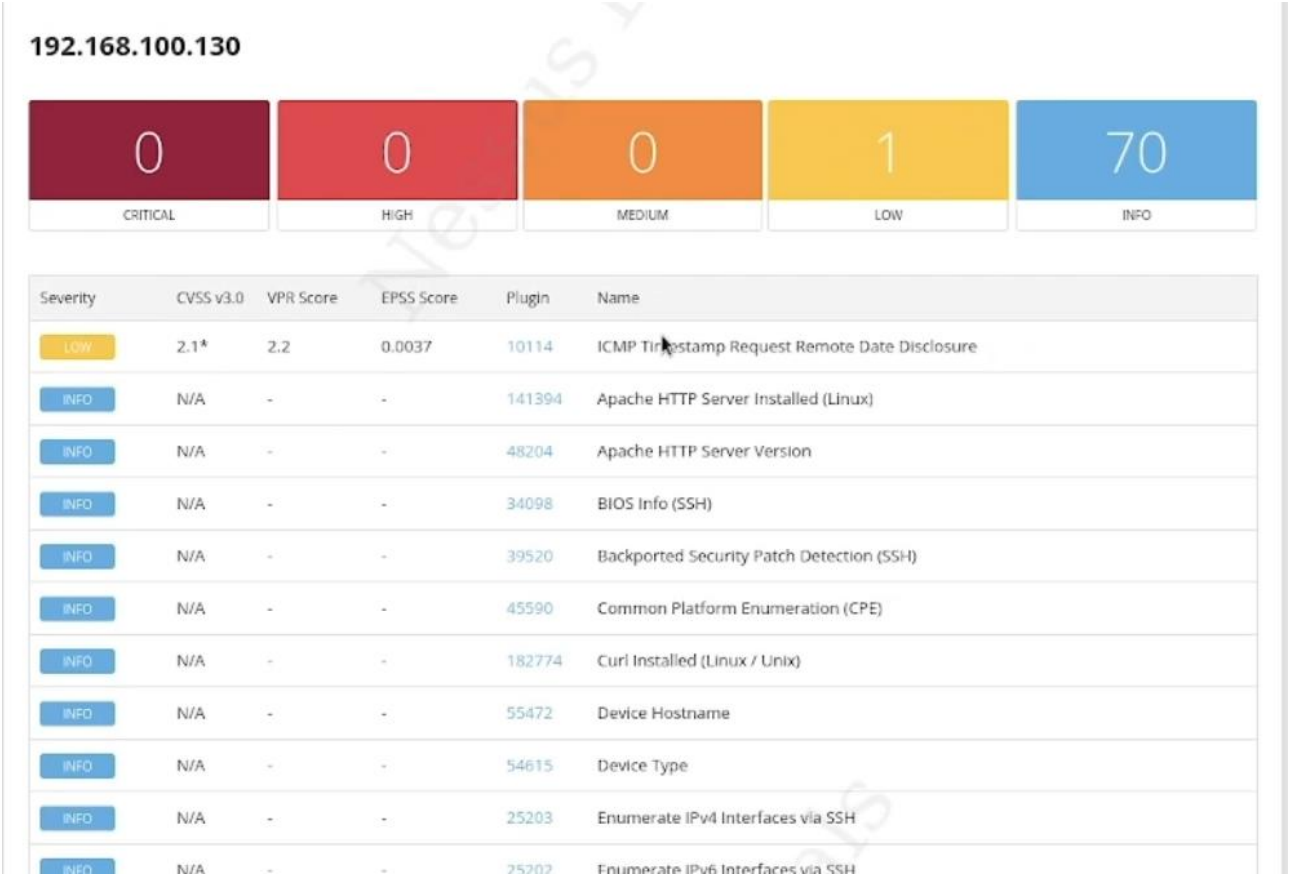
Nessus

Primer escaneo:



Segundo escaneo (Vulnerabilidades agregadas):

Tercer escaneo (Vulnerabilidades parchadas):



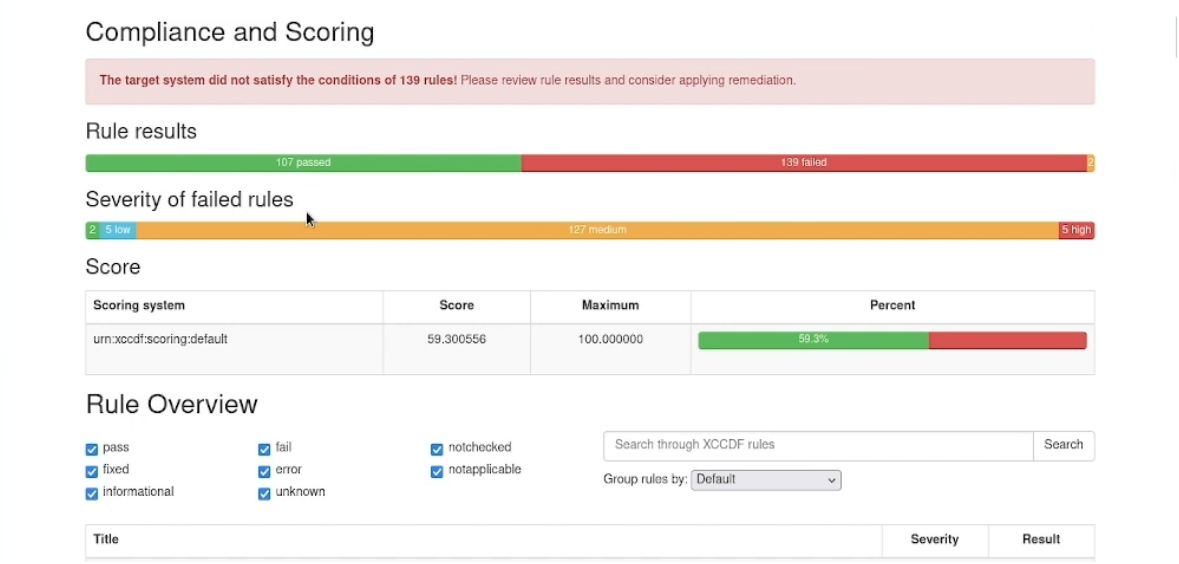
192.168.100.130

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	0	0	1	70

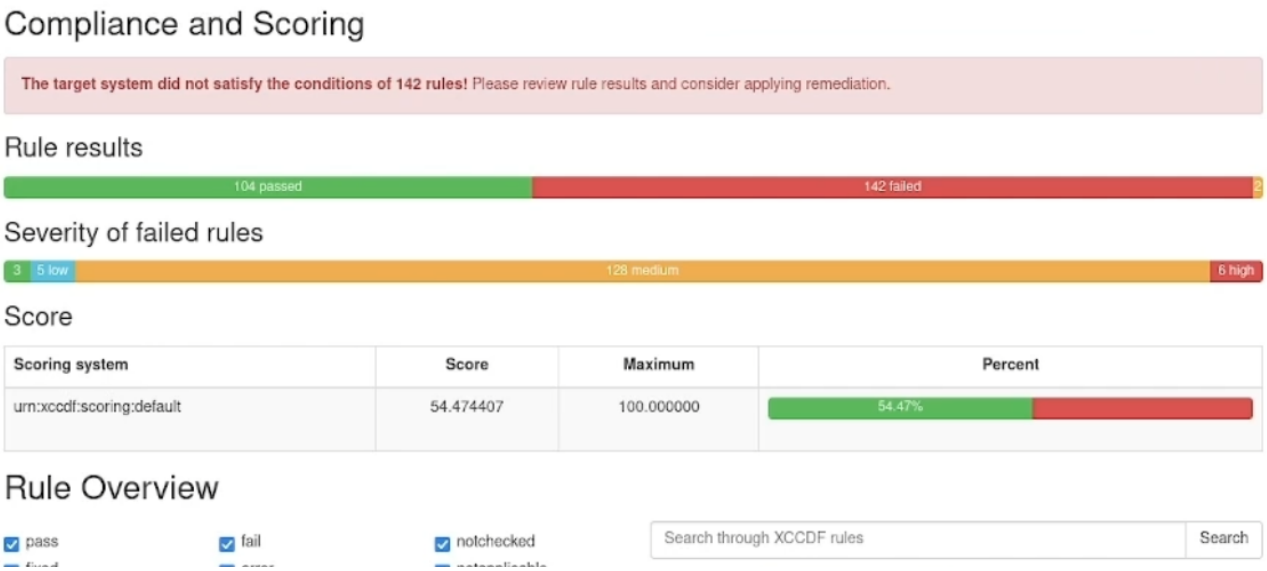
Severity	CVSS v3.0	VPR Score	EPSS Score	Plugin	Name
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	141394	Apache HTTP Server Installed (Linux)
INFO	N/A	-	-	48204	Apache HTTP Server Version
INFO	N/A	-	-	34098	BIOS Info (SSH)
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	182774	Curl Installed (Linux / Unix)
INFO	N/A	-	-	55472	Device Hostname
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	25203	Enumerate IPv4 Interfaces via SSH
INFO	N/A	-	-	25202	Enumerate IPv6 Interfaces via SSH

OpenScap

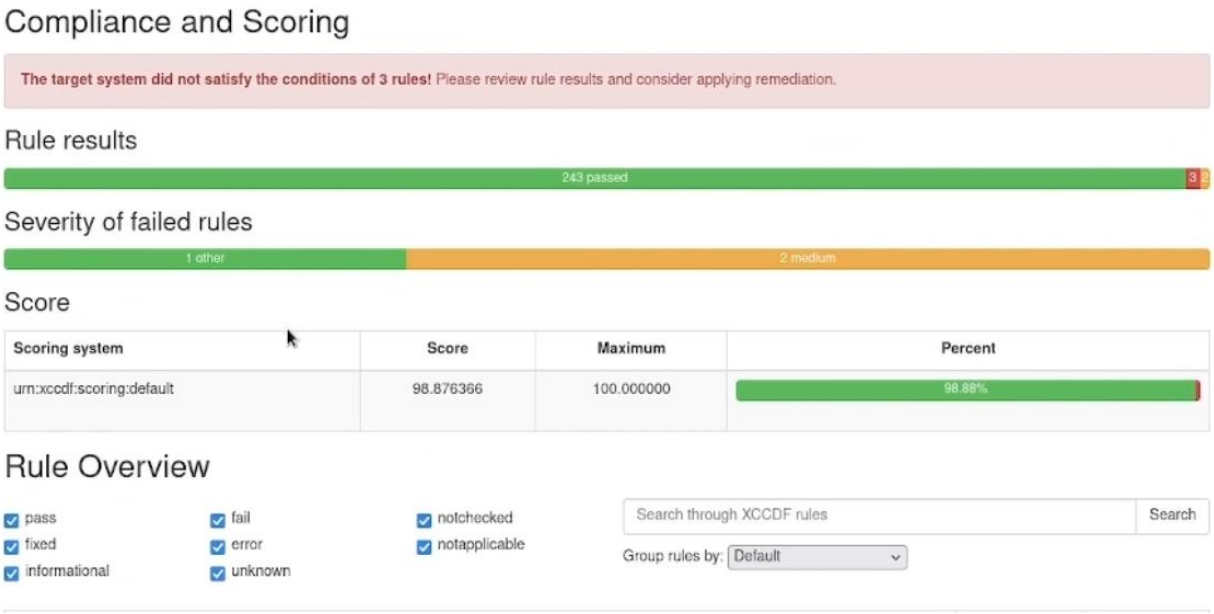
Primer Escaneo:



Segundo escaneo (Vulnerabilidades agregadas):



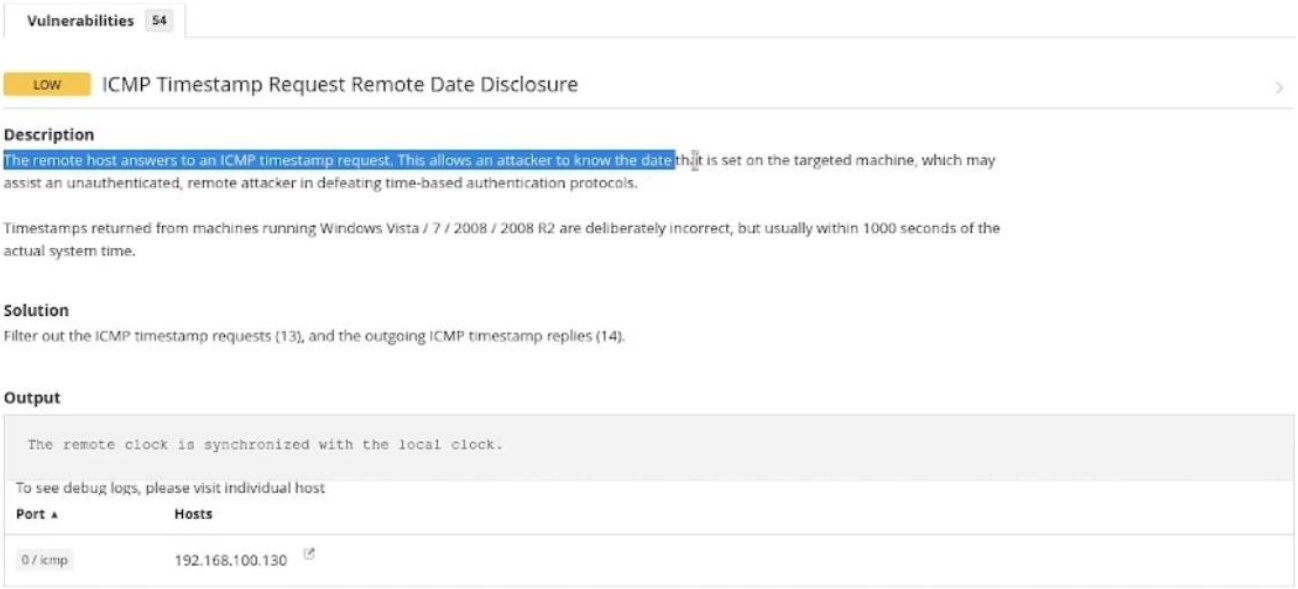
Tercer escaneo (Vulnerabilidades parchadas):



Aquí hemos visto como los escaneo y parchar las vulnerabilidades ha resultado un éxito, llegando prácticamente a mas del 95% de seguridad en el escaneo de OpenScap y dándonos una vista mucho más fácil de los errores que han sido corregidos.

Ahora debemos nosotros ver los errores que OpenScap no pudo resolver y veamos cuales son algunas soluciones dentro de lo que nos recomienda la herramientas y soluciones que podemos implementar nosotros mismos.

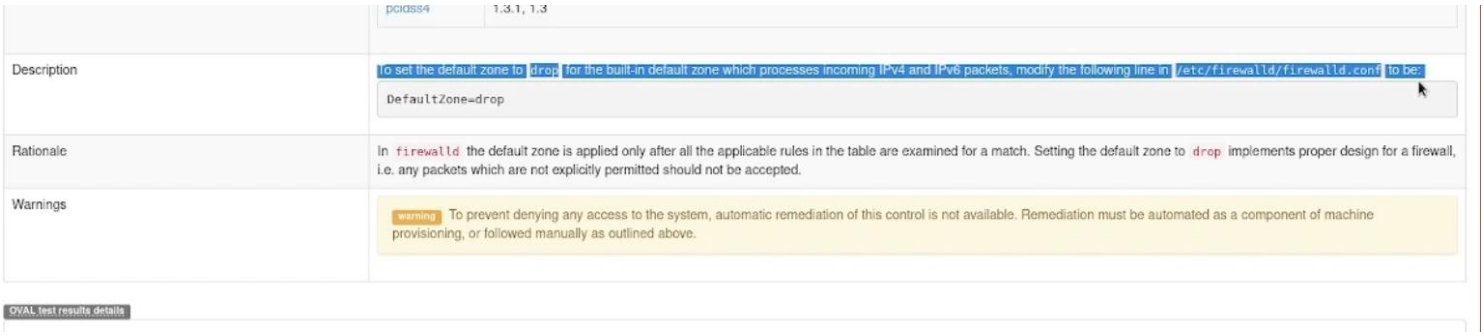
Primer error:



La solución de este error es simple ya que es más cuestión de ajustar el horario de la PC a un tiempo determinado dentro de 1000 segundos de la hora actual, esto debido a los métodos de autenticación basados en la hora, si lo tenemos en la hora actual lo detectara como un fallo de seguridad pequeño que podemos arreglar fácilmente.

OpenScap no lo arreglo ya que este es un problema que se presentó solamente en Nessus y que puede ser arreglado muy fácilmente

Segundo error:



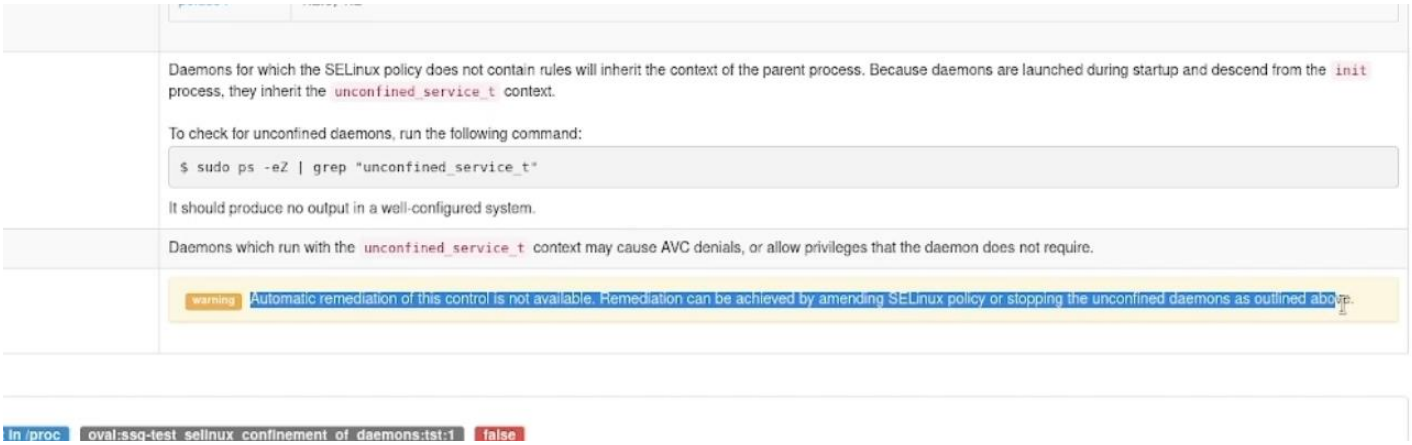
Este error tiene como nombre: Set default firewalld zone for incoming packets.

Este es un error que nos indica que tenemos que colocar una zona donde se soltaran los paquetes ipv4 e ipv6 de manera automática

Simplemente tenemos que entrar al archivo de configuración /etc/firewalld/firewalld.conf y en la sección de DefaultZone debemos poner: DefaultZone=Drop.

Como el mismo programa lo dice, este error no fue corregido por OpenScap para no causar interferencia con el usuario y no someter al sistema a que no pueda haber acceso a este, por eso el parcheo automatico esta desactivado para este error.

Tercer error:



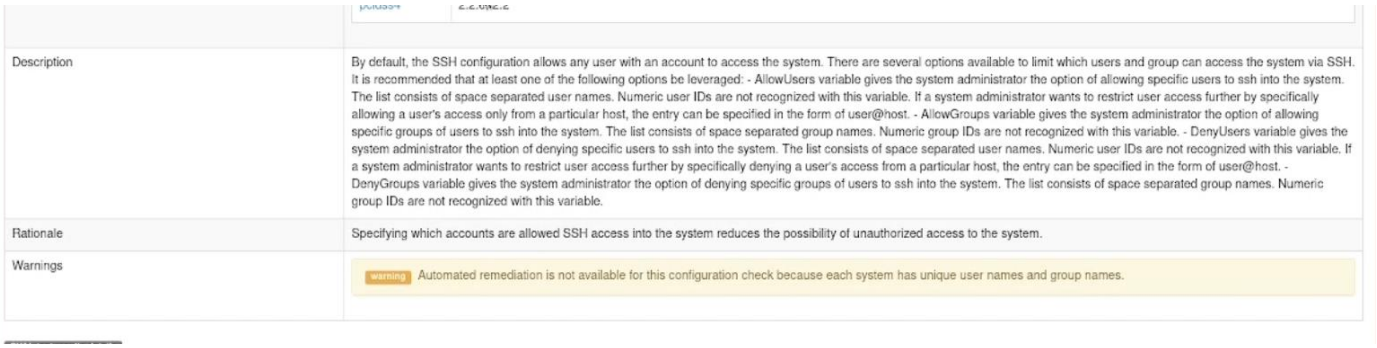
Este error tiene como nombre: Ensure no daemons are unconfined by SELinux, como explicamos antes nosotros hemos desactivado lo que son las opciones de SELinux para resguardar estos daemons importantes que tienen informaciones críticas del sistema.

Las soluciones que tenemos para este problema son las siguientes:

- Asegurar los daemons no confinado
- Para detectar los daemons (comando dado por OpenScap): sudo ps -eZ | grep "unconfined_service_t"
- Verificar estado de SELinux
- Comprueba que SELinux esté en modo enforcing usando: getenforce
- Si devuelve Permissive o Disabled, actívalo en /etc/selinux/config con: SELINUX=enforcing
- para aplicar los cambios: sudo setenforce 1

OpenScap no pudo resolver este error ya que este se encuentra fuera de las capacidades de este mismo, por ende se encuentra bloqueado

Cuarto error:



Description	By default, the SSH configuration allows any user with an account to access the system. There are several options available to limit which users and group can access the system via SSH. It is recommended that at least one of the following options be leveraged: - AllowUsers variable gives the system administrator the option of allowing specific users to ssh into the system. The list consists of space separated user names. Numeric user IDs are not recognized with this variable. If a system administrator wants to restrict user access further by specifically allowing a user's access only from a particular host, the entry can be specified in the form of user@host. - AllowGroups variable gives the system administrator the option of allowing specific groups of users to ssh into the system. The list consists of space separated group names. Numeric group IDs are not recognized with this variable. - DenyUsers variable gives the system administrator the option of denying specific users to ssh into the system. The list consists of space separated user names. Numeric user IDs are not recognized with this variable. If a system administrator wants to restrict user access further by specifically denying a user's access from a particular host, the entry can be specified in the form of user@host. - DenyGroups variable gives the system administrator the option of denying specific groups of users to ssh into the system. The list consists of space separated group names. Numeric group IDs are not recognized with this variable.
Rationale	Specifying which accounts are allowed SSH access into the system reduces the possibility of unauthorized access to the system.
Warnings	warning Automated remediation is not available for this configuration check because each system has unique user names and group names.

El ultimo error de nuestra lista tiene como nombre: Limit users ssh Access

Básicamente y resumiendo de gran manera lo que dice la descripción: no podemos dejar el puerto ssh desprotegido y con acceso para todo el público, dependiendo de que usuario tenga este puerto abierto puede considerarse amenaza aun mayor o menor pero no deja de ser amenaza ya que cualquiera podría meterse por el puerto y si logran crackear la contraseña sería mucho peor ya que debemos evitar eso a toda costa.

Algunas de las recomendaciones para nosotros cuidar el servicio ssh son las siguientes:

- Limitar qué usuarios pueden usar SSH
- sudo nano /etc/ssh/sshd_config
- Agrega al final: AllowUsers usuario1 usuario2
- Para denegar: DenyUsers usuario3 usuario4

- Limitar por direcciones IP:
- en /etc/ssh/sshd_config
- Permitir: AllowUsers usuario1@192.168.1.50 usuario2@10.0.0.*
- También se puede usar en /etc/hosts.allow:
- sshd: 192.168.1.50
- Y para negar todo lo demás en/etc/hosts.deny
- sshd: ALL

Por último, el servicio OpenScap no pudo parchar este error ya que esto es algo que debe ser determinado por el mismo usuario y seria incorrecto que lo haga el programa automáticamente ya que es el usuario que debe decidir quien o que pasa por ese puerto.

Conclusión

Dentro de este extenso trabajo final hemos sido testigos de muchas formas de poder descubrir vulnerabilidades, así como podemos usar los consejos de las herramientas dadas para la realización de este examen.

Podemos aprender cada vez mas de estos programas como lo son OpenScap que en mi caso me tomo por sorpresa por la efectividad de este a la hora de parchar los errores dentro del sistema y de eliminar incluso a hasta las cosas más minúsculas después de los análisis que este hizo dentro del sistema.

Pudimos ver también como los problemas fueron resueltos en tiempo real e incluso aunque alguno de ellos no haya sido posible de corregir, se nos dieron las recomendaciones suficientes para llevar a cabo nuestro cometido.

Sin ninguna duda la travesía a través de este examen fue muy curiosa, así como lo fue por la materia completa, fue un verdadero placer poder aprender más de esta forma y con las practicas necesarias que nos hicieron investigar de buena manera y aprender de sus contenidos para poder explicarlos en nuestros videos.

Sin más que agregar ni nada que solucionar me despido, ha sido un placer muy grande para mi dar mi esfuerzo en esta materia y mas con un profesor cuyas clases no pesaban, si no que se podían disfrutar, hasta aquí llega mi trabajo y si Dios quiere, nos veremos nuevamente mas adelante