

Utilizar Nessus para escanear las vulnerabilidades de su servidor

Ya teniendo listo nuestro **Nessus** le haremos un SCAN a nuestro servidor web para ver que resultados nos da sobre nuestro servidor y así de esta manera buscarle una solución a los errores mostrados. En el caso de este proyecto estaremos resolviendo solo los problemas (medium, high y critical).

En mi caso en el informe tuve hallazgos medios y high, los cuales fueron los siguientes. [Click aquí para ver el informe](#)

Ahora vamos a presentar la solución a cada uno de los problemas.

TLS Version 1.0 Protocol Detection

Este es un problema que se da mucho en los servidores web que no tienen un buen hardening implementado, y es causado debido a que el servicio remoto acepta conexiones cifradas mediante TLS 1.0. TLS 1.0 tiene una serie de defectos de diseño criptográfico. Las implementaciones modernas de TLS 1.0 mitigan estos problemas, pero las versiones más recientes de TLS, como la 1.2 y la 1.3, están diseñadas contra estos fallos y deberían utilizarse siempre que sea posible.

Solución

Simplemente tendremos que dirigirnos a nuestro archivo **ssl.conf** que en mi caso y seguramente en el de ustedes este archivo se encuentre en la ruta **/etc/httpd/conf.d/ssl.conf** abrimos este archivo en mi caso con nano

```
nano /etc/httpd/conf.d/ssl.conf
```

```
GNU nano 2.9.1 File: /etc/httpd/conf.d/ssl.conf
#
# When we also provide SSL we have to listen to the
# the HTTPS port in addition.
#
Listen 443 https

##
## SSL Global Context
##
## All SSL configuration in this context applies both to
## the main server and all SSL-enabled virtual hosts.
##

# Pass Phrase Dialog:
# Configure the pass phrase gathering process.
# The filtering dialog program ('builtin' is a internal
# terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog exec:/usr/libexec/httpd-ssl-pass-dialog

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use and second the expiring timeout (in seconds).
SSLSessionCache shmcb:/run/httpd/sslcache(512000)
SSLSessionCacheTimeout 300

# Pseudo Random Number Generator (PRNG):
# Configure one or more sources to seed the PRNG of the
# SSL library. The seed data should be of good random quality.
# WARNING! On some platforms /dev/random blocks if not enough entropy
# is available. This means you then cannot use the /dev/random device
# because it would lead to very long connection times (as long as
# it requires to make more entropy available). But usually, there
```