Trabajo de Fin de Máster

Máster en Ciberseguridad y privacidad

ASLENS

Análisis de Sistemas Linux para el Esquema Nacional de Seguridad

MEMORIA

7 de Septiembre de 2019

Autor: Raúl Leal Bachot

Tutor: Marta Beltrán Pardo

Co-tutor: Fernando Sevillano

Resumen

El objetivo de este proyecto es el desarrollo de una herramienta capaz de realizar un análisis del Esquema Nacional de Seguridad para distribuciones Linux debido a la carencia actual para este tipo de sistemas, ya que, actualmente solo se puede realizar el análisis técnico para Windows mediante la herramienta “CLARA”.

Para ello, se ha desarrollado una aplicación capaz de ejecutar dicho análisis, generando un reporte HTML y PDF, además de permitir realizar un análisis extra de diferentes componentes del sistema en función de un fichero JSON. Así mismo, debido a la cantidad de distribuciones existentes, se ha optado por realizar todas las pruebas en dos distribuciones conocidas, siendo estas las siguientes: Ubuntu y Red Hat.

Además de la compatibilidad entre diferentes distribuciones Linux, se ha optado por la compatibilidad entre versiones de Python, siendo estas las versiones 2.7.x y 3.x.

Índice

1. Introducción

1.1. Contexto

Actualmente en España nos encontramos con un gran número de administraciones públicas que día tras día siguen desarrollando su evolución hacia el mundo digital. Desde el año 2004 se pusieron en marcha numerosas iniciativas para impulsar el desarrollo de la administración electrónica. Con dicha evolución digital se impulsaron dos reales decretos (3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica y el Real Decreto 951/2015, de 23 de octubre, de modificación del Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica).

El objetivo de estos decretos era garantizar el correcto funcionamiento, así como la legitimidad de los sistemas a través de los medios electrónicos además de garantizar la aplicación segura de estas tecnologías. Por lo tanto, perseguían afianzar la confianza en los sistemas de información los cuales prestan sus servicios además de custodiar toda la información relevante a los individuos sin que se produzcan interrupciones o modificaciones no solicitadas de la información, teniendo en cuenta que estos sistemas se comunicarán con otros organismos públicos como empresas del sector privado.

Si además de esto, le unimos los constantes ataques informáticos contra los diferentes sistemas, el cumplimiento del ENS pretende aumentar la seguridad de estos además de minimizar la superficie de ataque, reduciendo de esta forma, los daños que puedan ser sufridos.

1.2. Partes Interesadas

Teniendo en cuenta el contexto arriba descrito, el ENS puede aplicarse a diferentes partes y actores:

- **Administraciones públicas**. Son un conjunto de organizaciones públicas que realizan la función administrativa y de gestión del Estado y de otros entes públicos con personalidad jurídica, ya sean de ámbito regional o local. Además, son las encargadas de velar por el correcto funcionamiento de sus sistemas informáticos junto al mantenimiento de un nivel de seguridad del ENS suficiente que garantice el correcto tratamiento de la información y minimice los riesgos que puedan afectar tanto a la administración pública como a los terceros.

  Asimismo, el ENS (que está estrechamente relacionado en la administración electrónica) y las administraciones públicas (que son las encargadas de la realización de la administración electrónica) están estrechamente relacionadas entre sí, por lo tanto, todas las administraciones públicas deben de aplicar el ENS.

- **Terceros**. Son los colectivos ya sea individuos físicos como organizaciones que se ven afectadas directamente por el correcto funcionamiento de la administración

pública ya que toda la información de estos es almacenada o manipulada por esta entidad pública. Por lo tanto, una correcta aplicación del ENS garantizaría los plenos derechos de los interesados.

- **Empresas privadas**. Aunque el Esquema Nacional de Seguridad nació pensado para las administraciones públicas, sí que pueden hacer uso de toda la documentación relacionado con el ENS y la herramienta ofrecida por el CNI como medida de aproximación al RGPD sin llegar a completar por completo dicho reglamento europeo

1.3. Aplicaciones, servicios o sistemas comprendidos en el ámbito de aplicación del ENS

Tal y como se recoge en toda la documentación ofrecida por el CNI, las aplicaciones que deben de aplicar el Esquema Nacional de Seguridad son las siguientes:

- **Sedes electrónicas**
- **Registros electrónicos**
- **Sistemas de información accesibles electrónicamente por los ciudadanos**
- **Sistemas de información para el ejercicio de derechos**
- **Sistemas de información para el cumplimiento de deberes**
- **Sistemas de información para recabar información y estado del procedimiento administrativo.**

Además, si hubiese un sistema que no se ampare en los indicados anteriormente, será necesario analizarlo en detalle para comprobar si debe de aplicar el ENS y en caso afirmativo, determinar el nivel correspondiente que debe de realizar.

1.4. Cumplimiento del Esquema Nacional de Seguridad

Para su cumplimiento, se detallan hasta 75 medidas de seguridad que pretenden recoger todos los principios básicos y/o requisitos mínimos aplicables a su dimensión de seguridad correspondiente como de la categoría del sistema de información.

Estas categorías son las siguientes:

- **Marco organizativo**: Medidas relacionadas con la organización global de la seguridad.
- **Marco operacional**: Medidas relacionadas para proteger la operación del sistema como conjunto integral de componentes para un fin.
- **Medidas de protección**: Medidas centradas en proteger activos concretos, ya sean por su naturaleza como por la calidad exigida por el nivel de seguridad de las dimensiones afectadas.

Un resumen de estas categorías junto a su correspondiente subconjunto puede ser resumido en la figura siguiente:

Figura 1: Medidas de Seguridad recogidas en el ENS. Analizando el anexo 2 denominado "Medidas de seguridad", recogido en el ENS, se puede analizar cada subconjunto de manera detalla. Fuente: https://www.liderit.es/wp-content/uploads/75_medidas_ENS.png

Siguiendo con la información aportada por el CNI, la adecuación al Esquema Nacional de Seguridad puede ser resumida en la siguiente figura:

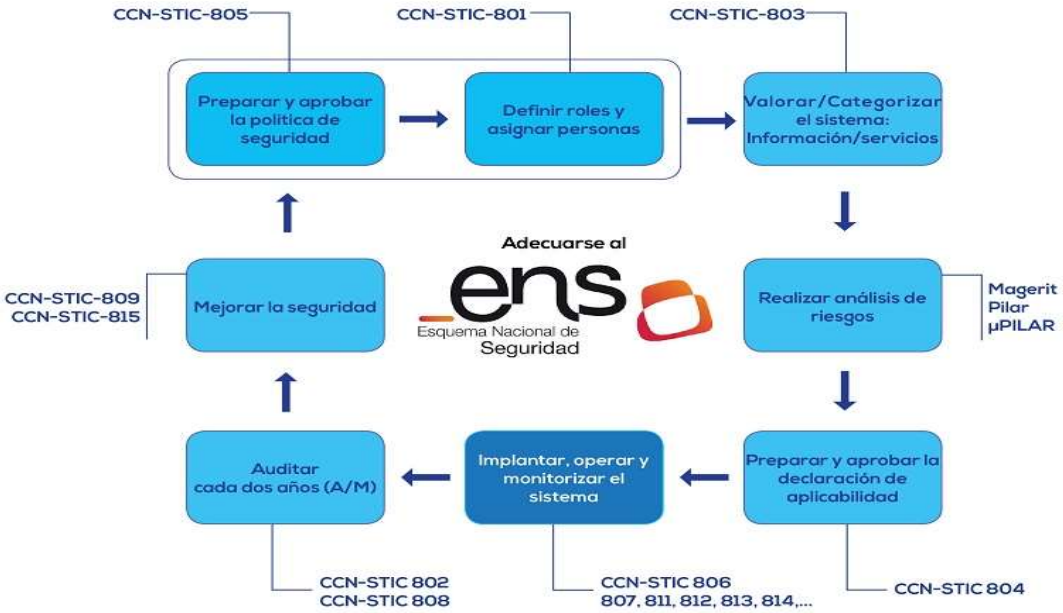

Figura 2: Adecuación al ENS. Fuente: https://www.ccn-cert.cni.es/images/grafico_adecuacion2.jpg

En esta figura, puede verse un diagrama para la adecuación al ENS, junto a normas, documentación y/o herramientas que facilitan la realización de este.

2. Estado del arte

Para diagnosticar si un determinado activo cumple con los requisitos expresados en el ENS, existen dos opciones: La verificación manual basándose en la guía CCN-STIC 808 y la verificación automática, basándose en la herramienta CLARA. Veámoslo:

- **Documento de Verificación del cumplimiento del ENS**. Este documento incluido en la categoría de Guía de Seguridad de las TIC (CCN-STIC 808), recoge una guía para realizar la auditoría de los requisitos del Esquema Nacional de Seguridad con independencia de la naturaleza y/o aplicación del sistema. Así mismo, por la manera en que dicho documento es generado, permite la utilización de este como herramienta de trabajo, ya que permite ir analizando punto a punto los diferentes requisitos del ENS, indicando si se aplica y audita además de indicar observaciones a la misma.

  Además de esto, permite definir claramente las diferentes categorías del requisito, que será denominado en este documento también como el nivel de seguridad del ENS mediante una serie de colores asignado los siguientes valores:
  - Categoría Básica - Verde
  - Categoría Media - Amarillo
  - Categoría Alta - Rojo

  Como información extra, esta guía añade información de donde puede ser consultada información sobre el requisito, utilizando todas las guías CCN-STIC de las que dispone el CNI.

- **Herramienta de seguridad (CLARA) desarrollada para analizar las características de seguridad técnicas del ENS**. Esta herramienta permite realizar un análisis técnico del sistema, pero será necesario la realización de una auditoría para detallar el resto de los requisitos no técnicos definidos en el ENS.

  CLARA permite seleccionar la categoría y/o nivel de seguridad que se va a aplicar y empezará a ejecutar un análisis del sistema para posteriormente generar un documento con formato HTML en el que se detalla lo siguiente:
  - Nivel de cumplimiento del ENS para el nivel indicado.
  - Porcentaje de este.
  - Listado con todos los requisitos desglosados, indicando si su valor es correcto o cuál es su valor esperado.
  - Información básica del sistema a título informativo.

  Por lo tanto, dicho documento puede ser utilizado como una base muy importante para realizar un análisis de seguridad de todo el ENS.

  Así mismo, tal y como es recogido en la web de descarga de la aplicación CLARA, sí que tiene un problema muy importante y es que es exclusivamente funcional para entornos Windows, tanto para versión cliente como para versión

servidor, para la mayoría de sus versiones (**Windows 7, Windows 10, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016**).

3. Objetivos

Tal y como se ha indicado anteriormente, este proyecto nace de la necesidad de disponer de una herramienta que realice el análisis técnico del ENS para entornos Linux.

3.1. Objetivos

El objetivo es desarrollar una herramienta que permita analizar el grado de cumplimiento de activos con SO Linux del ENS.

Como objetivos adicionales tenemos los siguientes:

- Estudiar la documentación existente para tomar la nota CCN-88 como punto de partida.
- Estudiar el análisis realizado por la aplicación CLARA como punto de partida para la parte técnica.
- Identificar la distribución más apropiada para un primer desarrollo
- Agrupar todo el análisis técnico en categorías o bloques, permitiendo así, una extrapolación correcta y liviana al proyecto.
- Generar un reporte lo suficientemente claro que permita la correcta lectura del ENS
- Utilizar una metodología de desarrollo de software seguro, en todo el código generado para esta herramienta.

3.2. Alcance

El alcance de este proyecto es realizar un documento, con el resultado el análisis además de incluir la información necesaria sobre el requisito, esclareciendo para que es dicho requisito o el valor que debe de contener. Así mismo, nos permite observar que, por defecto, los servidores Linux en los que se ha realizado la ejecución del proyecto, tienen un porcentaje muy bajo de requisitos del ENS satisfechos.

Se usará como base una distribución libre como Ubuntu para la prueba de este y una vez finalizado y comprobado que el análisis es correcto, se utilizará un sistema operativo de pago como RHEL, añadiendo al análisis las condiciones particulares para este sistema en ciertos ficheros, garantizando así que la herramienta sea compatible entre sistemas.

Por último, se ha incluido la posibilidad de incluir ficheros propios de configuraciones no incluidas en el ENS, permitiendo de esta forma, tener una herramienta abierta en cuanto a requisitos, de modo que servidores web como pueden ser Nginx o Tomcat, puedan ser incluidos en el análisis.

Aun así, en el desarrollo y/o ejecución de la herramienta, nos podemos encontrar las siguientes casuísticas que entorpecerían el correcto funcionamiento:

- **La gran cantidad de paquetes/librerías** pueden provocar que determinados requisitos no sean validados aun habiendo un software que realice esta función. Por ejemplo, un firewall o antivirus nuevo que no está incluido en los parámetros de control del software.

- **Estructura de ficheros diferente en función del tipo de sistema Linux** que se esté utilizando. No todas las versiones tienen la misma estructura/configuración de archivos, esto provoca que se tengan que añadir controles específicos para estas estructuras.
- Debido a la **gran cantidad de sistemas operativos Linux** que hay actualmente, realizar una herramienta que funcione para todas puede ser muy complicado, por ello, el propósito de esta herramienta es general, enfocándose en los principales servidores.

3.3. Barreras

Debido a la estructura de los SO a los cuales se ha enfocado esta herramienta, nos podemos encontrar con diferentes barreras que impedirán el correcto funcionamiento de la herramienta, como pueden ser las siguientes:

- **La gran cantidad de paquetes/librerías** pueden provocar que determinados requisitos no sean validados aun habiendo un software que realice esta función. Por ejemplo, un firewall o antivirus nuevo que no está incluido en los parámetros de control de la herramienta.

4. Planificación

4.1. Duración

La planificación del proyecto está prevista desde el 18 de marzo hasta el 5 de septiembre, ambos inclusive. El proyecto se inicia a partir de la asignación del tutor del proyecto hasta la entrega de la documentación y código de este.

4.2. Descripción de las tareas

La planificación del proyecto ha sido dividida en fases quedando definido de la siguiente forma:

Tareas	Horas
Inicio del TFM	**54**
Análisis de documentación del ENS	30
Búsqueda de información de CLARA	8
Análisis del reporte de CLARA	16
Desarrollo de herramienta ASLENS	**130**
Desarrollo de categoría organizativa	60
Desarrollo de categoría operacional	50
Desarrollo de categoría protección	20
Desarrollo inicial HTML	**20**
Desarrollo de versión inicial del reporte	20
Desarrollo de configuración extra	**25**
Desarrollo para analizar configuración extra mediante JSON	25
Predefensa	**10**
Preparación y realización de predefensa	10
Generación de HTML Avanzado y PDF	**36**
Mejora de plantilla HTML	10
Inclusión de información del sistema	10
Generación de documento PDF	16
Compatibilidad Multisistema y Multiplataforma	**30**
Desarrollo de compatibilidad para múltiples SO	20
Desarrollo de compatibilidad para versión de Python	10
Testing	**40**
Testing del sistema en varios SO	40
Documentación y refactorización del código	**36**
Refactorización del código	26
Documentación del código	10
Documentación y defensa	**21**
Preparación de memoria	15
Preparación de defensa	6
Total	**402**

Tabla 1: Horas estimadas del proyecto

5. Desarrollo

5.1. Arquitectura

La arquitectura utilizada para el desarrollo de la herramienta es la siguiente:

- Máquina Virtual (Virtual Box 6.0.10) con las siguientes características:
  - SO: Ubuntu 18.04.2 LTS 64 bits
  - Kernel: 4.15.0-58-generic
  - Microprocesador: Intel® Core™ i5-7300HQ CPU @ 2.50GHz × 4
  - RAM: 7.79 GB
- Python 3.6.8. Aunque para las pruebas finales y la interoperabilidad entre versiones de Python, se ha utilizado también la 2.7.15+
- Sublime Text 3 como IDE, junto a los siguientes plugins:
  - SublimeLinter
  - Trailing Spaces
  - Anaconda
  - Python PEP8

Como bien se ha indicado anteriormente, el sistema operativo base ha sido Ubuntu, pero debido a la naturaleza del proyecto, se ha necesito de otros SO Linux para comprobar que la funcionalidad desarrollada funciona correctamente. En este caso, se ha escogido también un SO basado en arquitectura RedHat.

5.2. Principales hitos

El desarrollo ha ido siempre acompañado a unos hitos/entregables, por lo tanto, todo desarrollo irá en función de estos entregables.

5.2.1. Elaboración de documento HTML con los resultados del análisis

Este hito es el más importante, ya que incluye la realización de todo el análisis de lo que es un Esquema Nacional de Seguridad y su extrapolación a sistemas Ubuntu, siguiendo como base el documento citado en el "Estado del arte" de este documento o la propia aplicación "Clara", teniendo en cuenta que muchos análisis efectuados por esta herramienta, se basan en los registros de Windows.

Una vez realizado ya el análisis correspondiente y teniendo una idea clara del objetivo final, lo siguiente que se realizó en el desarrollo de este, son las agrupaciones de los requisitos en una serie de categorías que garantizasen que dicha información se podía obtener de forma técnica sin intervención humana.

Antes de continuar con las agrupaciones, cada una de ellas se compone de varios subconjuntos, que se irán detallando a continuación. Así mismo cada agrupación se compone de varios subconjuntos, que se irán detallando a continuación a lo largo de este hito. Debido a la naturaleza que presentan todas estas agrupaciones y sus subconjuntos, para aprovechar la funcionalidad del código y para mantener una limpieza y orden de este, se ha optado por utilizar un modelo de herencia en el cual, todos los subconjuntos de las agrupaciones anteriores heredan de la misma clase.

Esto permite, que todos los cambios que haya realizar por cada clase o subconjunto puedan ser centralizados en un mismo sitio. Una vez realizada esta aclaración, lo siguiente es ir definiendo todos subconjuntos pertenecientes a las agrupaciones indicadas.

Las agrupaciones son las siguientes:

Marco Operacional – Control de acceso (op.acc)

Esta agrupación, está destinada principalmente a determinar y controlar el acceso a determinados elementos del sistema. Para ello, en el documento proporcionado por el CNI, establece una serie características que se deben de cumplir. En el proyecto desarrollado, para esta agrupación, se realiza los siguientes análisis:

- **Proceso de gestión de los derechos de acceso**. Para este análisis, se comprueban las propiedades relacionadas con el uso del super usuario (sudo), en este caso, comprobar si se pregunta la contraseña sudo y cada cuanto tiempo se vuelve a preguntar la misma. Como ayuda a esto, en el reporte generado ya sea HTML o PDF, indica una serie de información para un mejor entendimiento.
- **Mecanismos de autenticación**. Para este análisis, se comprueban las propiedades relacionadas con la contraseña de los usuarios, ya sea complejidad, tamaño mínimo, número de contraseñas para repetir la misma, días entre los cuales puede cambiar la contraseña, además de otras propiedades. Así mismo, se analiza cuando días deben de pasar para bloquear cuentas inactivas, se comprueba si existen usuarios sin contraseñas o si se hace uso de FIPS, estándar de seguridad de los ordenadores de Estados Unidos para la acreditación de módulos criptográficos.
- **Acceso local**. Para este análisis, se comprueban los parámetros o propiedades enfocados al acceso local de los usuarios al sistema. Para ello, se comprueba el número de intentos erróneos hasta producirse el bloqueo de la cuenta, restablecer las cuentas bloqueadas después de un tiempo determinado, bloquear la cuenta root si se producen un número determinado de fallos o comprobar si existe en el sistema, más de un super usuario.
- **Acceso remoto**. Para este último análisis, se comprueban todos los parámetros de esta agrupación. Este acceso remoto, en esta herramienta, es únicamente para SSH, por lo que es necesario que en el sistema exista un servidor SSH. Las propiedades que se comprueban, si que varían un poco respecto a todo lo visto anteriormente, pero es debido a que se pueden producir accesos desde cualquier parte. Para ello se comprueba si se permite acceso root remotamente, si está restringido el acceso a determinadas IP's, si está restringido solo a ciertos usuarios, si utiliza un protocolo adecuado, unas serie de políticas de seguridad para el SSH como puede ser deshabilitar RHosts y el X11Forwarding, permitir contraseñas vacías, limitar un número máximo de intentos de sesión, si usa PAM para la política de contraseñas y acceso, si la autenticación es mediante clave pública/privada o el mostrar información del último inicio de sesión realizado.

Como se puede ver, hay muchísima más gestión para esta agrupación que para el resto.

Con estos 4 análisis, ya estarían realizado el análisis para el control de acceso. Tal como está desarrollada la herramienta, permite que mediante un fichero JSON se puedan modificar ciertas restricciones, como puede ser modificar el tamaño mínimo de la contraseña, el número mínimo de días inactivo entre otras propiedades. Para ello, se puede consultar el anexo de los ficheros adicionales de configuración.

Marco Operacional – Explotación (op.exp)

Esta agrupación está directamente enfocada a la explotación de los sistemas tal y como indica su nombre. Siguiendo la guía indicada a lo largo de todo el proyecto, se han determinado una serie de análisis que se van a ir detallando a continuación:

- **Configuraciones de seguridad**. Este análisis es el más largo y abstracto de todos los que se han implementado. Esto es debido a que existen muchos parámetros que permiten aumentar la seguridad del sistema, como puede ser la protección del boot, evitar cargar memorias USB, protección de vulnerabilidades TOCTOU, protección ExecShield activa, protección frente a mensajes malformados, protección syn-flood activada, entre otras muchas configuraciones del sistema. Actualmente está enfocado prácticamente para protección contra ataques de red, pero analizando diferentes ficheros se puede añadir muchas más propiedades de los sistemas Linux.
- **Gestión de cambios**. Este análisis está muy enfocado a las actualizaciones del sistema, ya que por seguridad todos los sistemas deben de estar actualizados además de comprobar si hubiese algún firewall activo. Aquí puede entrar en conflicto que el sistema tenga un firewall activado, pero no que no se encuentre en el listado, pero, si que la modificación del fichero JSON citado anteriormente, nos permite añadir más firewalls.
- **Protección frente a código dañino**. Para este análisis se analiza única y exclusivamente si hubiese un antivirus en el sistema. Para este, aplicaría lo citado anteriormente para los firewalls.
- **Registro de actividades y protección de estos**. En este análisis, se analizan una serie de logs del sistema, más específicamente los siguientes:
    - Log del kernel
    - Log de login en el sistema
    - Log del sistema
    - Log de arranque del sistema
    - Log de autenticación y privilegios
    - Log de conexión y desconexión de usuarios

  Para todos estos logs, se determina si existen y además se comprueba los permisos de los ficheros, comprobando si existen usuarios que pueden leer o modificar el fichero sin que debieran de hacerlo.

Con esto, ya se tendría analizado esta agrupación. La siguiente agrupación ya se corresponde con un conjunto diferente, ya que se pasará del marco operacional a las medidas de protección.

Medidas de protección

En esta última agrupación, nos encontramos ante la protección de los activos. Estos activos son el puesto de trabajo, más concretamente el bloqueo de este, la protección de los equipos, específicamente en el uso de particiones cifradas o la detección de servicios del sistema inseguros que van en contra de la autenticidad y de la integridad tanto de los usuarios como del propio sistema. Para ello, de la misma manera que los análisis anteriores, se dispone de varios análisis:

- **Bloqueo del puesto de trabajo**. En este tipo de análisis se determina el tiempo que debe de ocurrir sin actividad para que se produzca un bloqueo del puesto de trabajo o de la sesión de este si nos encontramos ante una conexión remota. Este tiempo está definido entre 5 y 10 minutos, pero es modificable desde el fichero JSON de configuración.
- **Protección de los equipos informáticos**. Para este análisis, se determina si el sistema está haciendo uso de particiones cifradas, para la protección de la información que tengan dichos equipos. Así mismo, se determina si la partición SWAP está cifrada y si el cifrado utilizado es lo suficientemente seguro. Junto a todo esto y como medida de protección adicional, se comprueba si los puntos de montaje del sistema operativo se encuentran en diferentes particiones separadas entre sí.
- **Protección de la autenticidad y de la integridad**. Hay determinados servicios muy comunes que están considerados inseguros. Estos son los siguientes:

  Telnet, Ftp, Tftp, RLogin, Rsh, Nis, Talk

  Todos estos servicios deberían de ser eliminados del sistema, ya que pueden provocar graves fallos contra la autenticidad y la integridad del sistema.

Una vez finalizado todos los análisis indicados anteriormente, estos han de ser exportados a un fichero para que estos datos puedan ser legibles y sirva como justificante del análisis realizado. En este primer hito, el fichero HTML generado era muy minimalista en el que mostraba toda la información agrupada y permitía identificar mediante colores, los análisis que fueron validados satisfactoriamente y aquellos que no, quedando un resultado similar al siguiente:

OP.EXP.5 - Gestión de Cambios		
Entrada	**Notas**	**Resultado**
Actualizaciones del sistema	Hay 172 actualizaciones pendientes	Incorrecto
Actualizaciones de seguridad	No hay actualizaciones de seguridad pendientes	Correcto
Firewall del sistema	Se ha detectado los siguientes firewalls activos: ufw	Correcto
OP.EXP.6 - Protección frente a código dañino		
Entrada	**Notas**	**Resultado**
Antivirus	No hay ningún antivirus activo	Incorrecto

Figura 3: HTML Hito 1

Una vez finalizado este hito, el siguiente hito claramente iba destinado a mejorar el documento HTML generado y a generar un documento PDF partiendo como base del HTML anterior. Así mismo, una vez hecha esta base de todo el código, por cada hito se va haciendo una refactorización del código o mejoras de este.

5.2.2. Mejora del documento HTML, generación del PDF y refactorización y reestructuración del código

Como se ha indicado, este hito está claramente enfocado a la información que mostrará al usuario una vez realizado todo el análisis. Como se pudo ver en la figura anterior, esta versión presentaba poca legibilidad o no incluía una información suficiente.

Para ello, añadiendo una serie de modificaciones, tanto en la plantilla como en el propio código de la herramienta, sí que se produce un gran cambio importante en lo que a información se refiere, quedando tal que así:

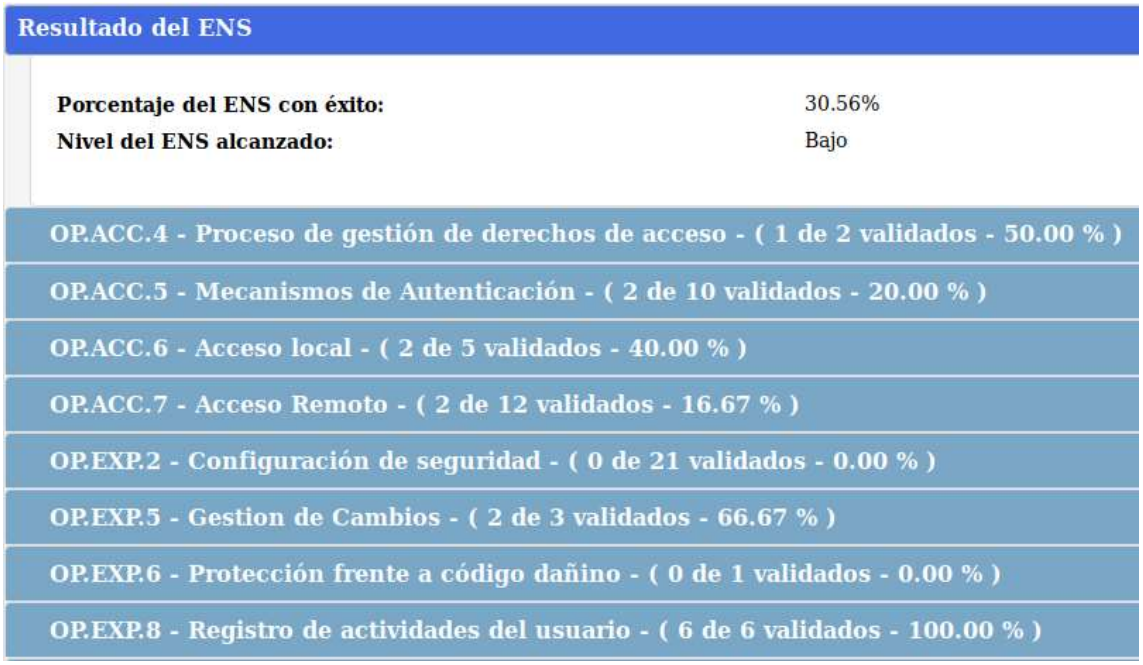

Figura 4: Resumen Análisis HTML

En este caso en cuanto a apariencia se refiere, sí que se ha producido un cambio importante. Pero ahora además hay un primer apartado, siendo este un resumen total del Esquema Nacional de Seguridad, indicando el porcentaje total y el nivel de seguridad alcanzado.

Además, ahora por cada análisis indica, cuantas propiedades se han comprobado correctamente del total a comprobar para dicho análisis. Junto a esto, se le indica también el porcentaje.

Junto a todo esto, para ofrecer más información del sistema que se está analizando y una mejor apariencia, se incluyó un encabezado que ofrece una serie de información básica, que se corresponde con información del sistema operativo, del

kernel, de la memoria, del uso del HD e información de las interfaces de red. Dicha cabecera, tiene una apariencia como la siguiente figura:

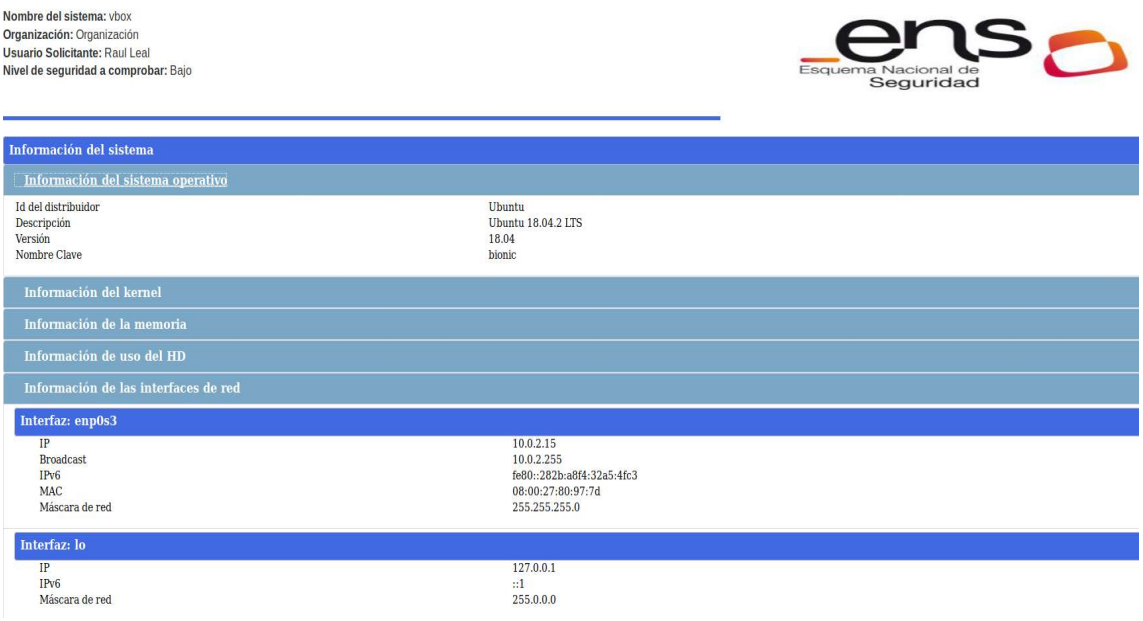

Figura 5: Cabecera del documento HTML

Por lo tanto, como se puede comprobar con las dos figuras anteriores y se ha indicado, el cambio en el HTML ha sido notorio.

Una vez realizado esto, lo siguiente es la generación de un documento PDF a partir del documento generado previamente. Esta generación es opcional y dependerá de la ejecución que se pretenda realizar de la herramienta.

Por último, en este hito se ha modificado la estructura del código pasando a ser una estructura arbórea y no una estructura plana, permitiendo así una mayor comprensión del código y una mejor visualización de este.

Finalmente, la realización de todos los puntos indicados previamente se puede dar por finalizado el hito y proceder a realizar el siguiente hito, siendo uno de los más importantes, ya que realización del Hito 3, garantiza que la herramienta desarrollada sea compatible para diferentes sistemas operativos Linux.

5.2.3. Compatibilidad con sistemas RHEL, documentación del código y configuración extra

Teniendo ya un documento HTML correcto y del cual se puede generar un documento PDF, lo siguiente a realizar es utilizar es utilizar esta misma herramienta, pero ya no en un sistema Ubuntu, sino en un Red Hat.

Partiendo de la base de que los sistemas RHEL son sistemas de pago, se ha obtenido una licencia de prueba de 1 mes para la realización de este hito. Una vez instalado el sistema operativo, lo siguiente a realizar es la ejecución del código.

En este caso, lo único problemático, surgía en el tratamiento de los ficheros PAM, que los nombres de los ficheros son renombrados de Ubuntu a Red Hat. A excepción de

esto ningún otro fichero produjo conflictos, por lo que tras una serie de modificaciones la adaptación fue todo un éxito.

Junto a todo esto, se ha desarrollado además una funcionalidad en la que mediante un fichero JSON externo (consultar Anexo 4 para más información), se pueden realizar análisis de diferentes componentes del sistema operativo, como puede ser configuraciones de un servidor Nginx o un servidor tomcat. Además, estos resultados, son agregados al HTML correspondiente, creando una nueva pestaña específica para este apartado.

Figura 6: Resultado de configuración extra en HTML

Una vez realizada la adaptación, lo siguiente a realizar en este hito igual que se han ido haciendo en todos los anteriores se corresponde con una limpieza y análisis de todo el código. Además, se ha realizado la documentación de todas las funciones utilizando el formato “napoleon” para realizar los docstrings.

Además de esto, se han utilizado herramientas de código estático para Python, para minimizar el número de errores, aplicando así una metodología de software seguro. Este hito ya finalizado, sería el último hito justo antes de la entrega final, por lo que lo único que quedaría para el hito final es la limpieza y toda la mejora posible del código.

5.2.4. Optimización y documentación del código

Tal y como se ha indicado con anterioridad, en este hito, se pretende realizar una optimización del código y la realización de toda la documentación que quedase pendiente. De esto, si que hay que destacar que en dicha optimización se ha realizado una compatibilidad entre diferentes versiones de Python, permitiendo ejecutar la aplicación con Python 2.7 y con Python 3.

Una vez hecho esto, solo queda realizar el último commit al repositorio y se daría el código por finalizado.

5.3. Implementación

La implementación de la herramienta se basa en la ejecución del código utilizando Python. Es importante aclarar, que debido a la naturaleza de los ficheros que se van a consultar, es necesario ejecutarlo como super usuario. En caso contrario saltará un aviso informando de que debe de ejecutar administrador.

El comando necesario para ejecutar el proyecto puede tener un aspecto como la siguiente figura:

Figura 7: Ejecución del proyecto

6. Conclusiones

En este TFM se ha procedido a desarrollar una herramienta que permite realizar el ENS para SO Linux utilizando como base la información proporcionada por el CNI además de una herramienta existen desarrollada por ellos mismos, pero con la exclusividad para entornos Windows. Así mismo, esta herramienta se ha desarrollado para el uso en diferentes SO, pretendiéndose la no exclusividad entre las diferentes distribuciones.

De tal forma que todas las pruebas realizadas a lo largo de todo el desarrollado ha permitido generar informes del ENS correctos para distribuciones Ubuntu (18 y 16) así como de distribuciones RHEL (8).

Dicho reporte generado por la herramienta pretende dar una serie de información básica sobre el SO actual sobre el que se está aplicando y los resultados de todo el ENS, ofreciendo todos los detalles de manera entendible y simple.

Además de realizar la compatibilidad entre diferentes distribuciones Linux, se ha realizado la compatibilidad entre diferentes versiones de Python, permitiendo así más compatibilidad entre las diferentes versiones del lenguaje.

Otro detalle a tener en cuenta es que este proyecto es capaz de generar un documento PDF con el resultado del análisis partiendo del HTML, pero la aplicación "CLARA", del CNI, solo es capaz de generar un HTML, sin tener en cuenta la exclusividad que tiene esta aplicación para sistemas Windows.

También, a lo largo de todo el desarrollo, se ha pretendido utilizar una metodología de desarrollo de software seguro, por lo que a medida que se iba realizando código, se han utilizado herramientas de validación de código, pretendiendo simular la función de un secdevops.

7. Futuras ampliaciones

A continuación, se enumerarán las diferentes ampliaciones que se podrían realizar a este proyecto, para que pudiese convertirse en una herramienta importante, sustituyendo incluso a la aplicación “CLARA”.

- Hacer el proyecto compatible para sistemas Linux también, de esta forma se sustituiría a la aplicación original, ya que, usando la librería de Python para consultar registros de Windows, se podría llevar a cabo gran parte de la funcionalidad de “CLARA”.
- Seguir ampliando análisis a realizar en los diferentes sistemas.
- Seguir probando la herramienta con múltiples sistemas operativos Linux, además de Ubuntu y RHEL.
- Usar también esta herramienta para realizar análisis de seguridad de los sistemas operativos, generando en el informe HTML, un listado de vulnerabilidades conocidas permitiendo así detectar estas vulnerabilidades al aplicar el ENS.

8. Tabla de ilustraciones

9. Bibliografía

1. https://www.ccn-cert.cni.es/ens.html
2. https://www.ccn-cert.cni.es/publico/dmpublidocuments/ENS-FAQ.pdf
3. https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/518-ccn-stic-808-verificacion-del-cumplimiento-de-las-medidas-en-el-ens-borrador/file.html
4. https://docplayer.es/11612199-Esquema-nacional-de-seguridad-ens-estado-de-situacion-y-retos-proximos.html
5. https://www.ccn-cert.cni.es/publico/ens/ens/index.html#!1010
6. https://www.ccn-cert.cni.es/soluciones-seguridad/clara.html
7. Google como medio de búsqueda para obtener información de los ficheros del entorno Linux

10. Anexo 1: Diagrama de GANTT

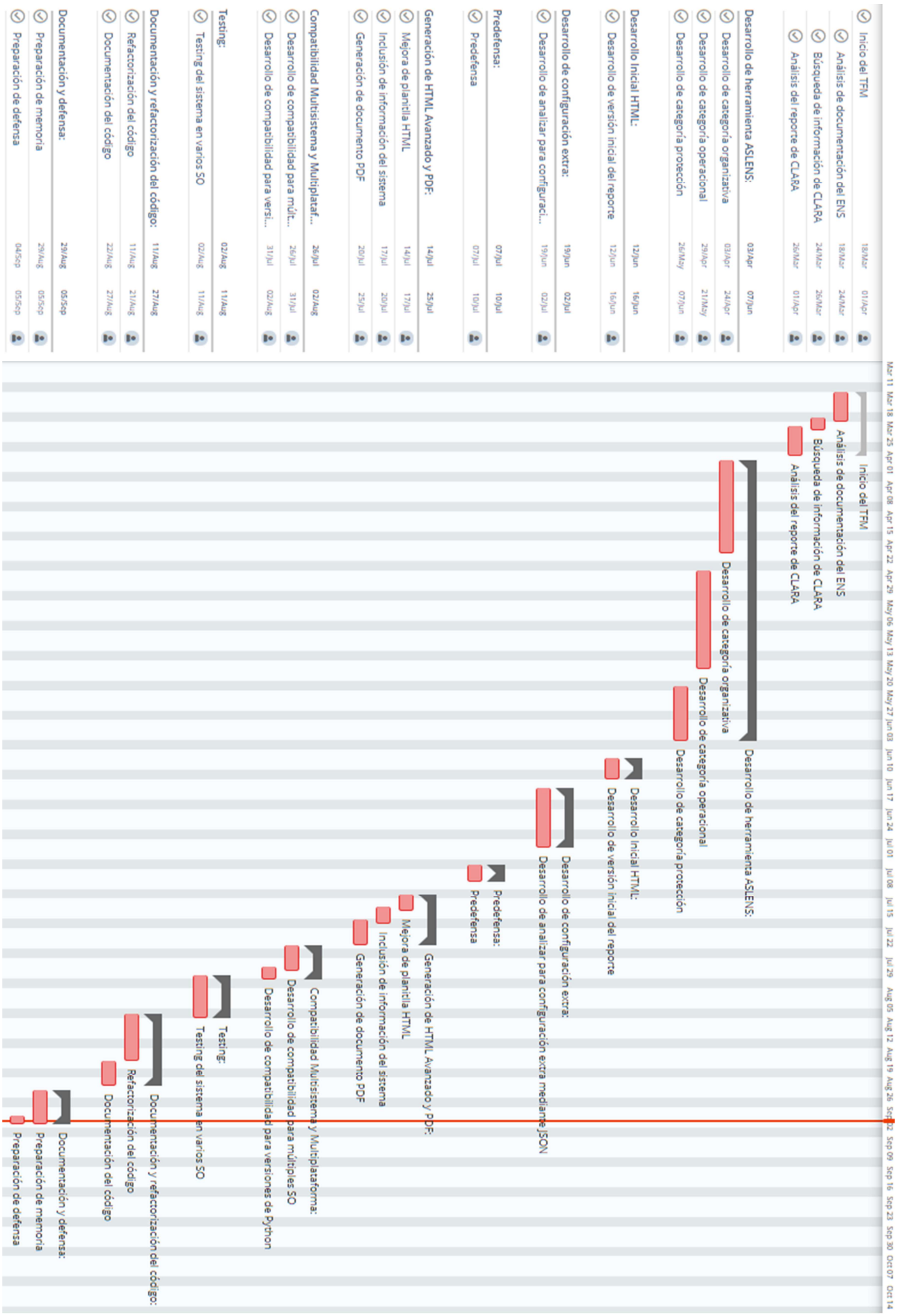

Figura 8: Diagrama GANTT

11. Anexo 2: Estructuración del código

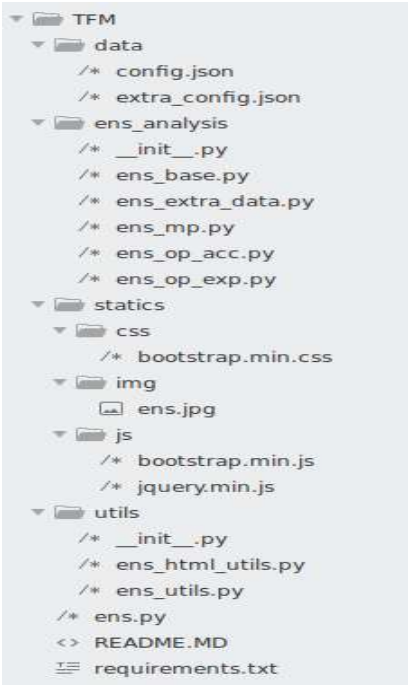

Figura 9: Estructura del código

12. Anexo 3: Agregación de análisis

En caso de querer añadir nuevos análisis a los diferentes conjuntos citados en el punto 5.2.1 de este documento, se debe de seguir la siguiente estructura:

- **Generación del método** correspondiente dentro del conjunto/clase, siguiendo la siguiente estructura:

```
def method_name(self):
    """
        Write here your code
```

Una vez defina esta estructura, el método debe de acabar de la siguiente forma:

```
self.entries_to_display.append(    [
        Analysis name (str),
        Value Found in file or default,
        Result (str, only can be 'Correcto' or 'Incorrecto')
        Description (str, optional)
    ]
```

El resultado de añadir esto último, tiene como objetivo generar la información necesaria para mostrar el documento HTML correctamente.

- **Clasificación del nuevo método**. La clasificación del método generado permite identificar al sistema si esta nueva función está asociada a un nivel de seguridad bajo, medio o alto. Además, esta clasificación es obligatoria debido a que si este método no se encuentra en alguna de estas categorías no será ejecutado.

Para ello, en la función “get_params”, disponible en todas las clases, hay que añadir la nueva función a alguna de las claves del diccionario “config”, siendo el valor de las claves las siguientes:

- **0**, Nivel de seguridad bajo
- **1,** Nivel de seguridad medio
- **2,** Nivel de seguridad alto

El resultado será similar a lo siguiente:

```
def get_params(self):
    ...
    config = {
        0: [
            ...,
            self.method_name,
        ],
        1: [],
        2: [],
    }
    ...
```

De esta manera, se ejecutará dicha función cuando el nivel de seguridad sea bajo o superior.

13. Anexo 4: Configuración extra

Como se ha comentado varias veces a lo largo de este documento, mediante un fichero JSON, se puede incluir análisis extra en el reporte HTML. Dicho fichero debe de seguir la siguiente estructura:

```
{
	group_name: [
		[path, property, ignore_lines_start_with_a_character,
property_separator, name_in_html, expected_value]
	],
}

group_name: (str, name to group the analyzes)
path: (str, file path)
property: (str, property in file)
ignore_lines_start_with_a_character: (str, ignore lines start with a
character as #)
property_separator: (str, character to split property of its value as
=)
name_in_html: (str, name in html)
expected_value: (str, expected value for the indicated property)
```

Un ejemplo de uso puede ser lo siguiente:

```
{
	"tomcat": [
		["/etc/tomcat8/logging.properties",
"org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level",
"#", "=", "Nivel de log del tomcat", "INFO"]
	],
	"nginx": [
		["/etc/nginx/nginx.conf", "server_tokens", "#", " ",
"¿Todos los servers tienen el token oculto?", "off"]
	]
}
```

El resultado de este fichero puede ser consultado en la figura 6.