Securización de infraestructura de red mediante OpenSCAP

Santiago Sierra González

Resumen– La gestión de la seguridad es un aspecto que cada vez gana más peso en un mundo con un crecimiento constante de los sistemas de información y las infraestructuras de red. Este proyecto pretende explorar y poner en práctica el protocolo SCAP con el objetivo de sentar una base introductoria para posteriores proyectos relacionados con la securización de infraestructuras de red. Este proyecto se centra en SCAP, un protocolo para la estandarización de la seguridad de sistemas y aplicaciones. Este trabajo aborda que es SCAP, sus componentes y funcionamiento sobre el papel y muestra un caso práctico sencillo diseñado específicamente para mostrar un ejemplo de securización de una infraestructura de red usando dicho protocolo.

Palabras clave– Checklist, SCAP, política de seguridad, OpenSCAP, open source y vulnerabilidad.

Abstract– Security management is an aspect that is gaining more and more weight in a world with a constant growth of information systems and network infrastructures. This project aims to explore and implement the SCAP protocol in order to establish an introductory basis for subsequent projects related to the securing of network infrastructures. This project is focused in SCAP, a protocol for the standarization of system and applications security. This project abords what is SCAP, its components and operation on paper and shows a simple practical case specifically designed to show an example of securing a network infrastructure using said protocol.

Keywords– Checklist, SCAP, security policy, OpenSCAP, open source and vulnerability.

1 Introducción

La cantidad de usuarios y sistemas crece cada vez más y esto lleva a unas infraestructuras de red, tanto profesionales como personales, crecientes en tamaño y complejidad. En la figura 1 se puede observar la cantidad de usuarios de internet a nivel mundial.

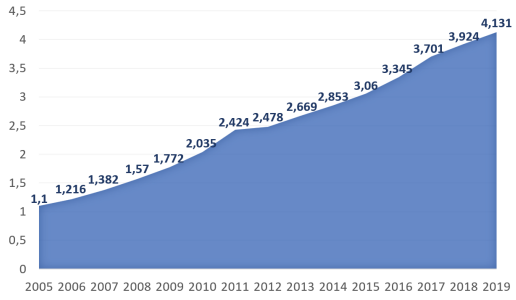

Fig. 1: Número de usuarios de Internet mundialmente (en millones).[1]

Esto dificultad la gestión de la seguridad de dichos sistemas, pues cuanto más abarca un sistema más propenso es a tener una vulnerabilidad. Esta situación se agrava con la tendencia que se observa desde hace años con respecto a la creciente cantidad de incidentes de ciberseguridad, como se puede apreciar en la figura 2.

Estas dos principales situaciones derivan en una necesidad creciente de implementar seguridad en los sistemas e infraestructuras de red. Una necesidad que a menudo se gestiona de forma muy heterogénea en cuanto a la decisión, denominación y gestión de la seguridad. Esta heterogeneidad puede llevar a ineficiencias o a problemas de entendimiento entre sistemas. Es una situación similar al intentar escribir un documento en varios idiomas distintos, no todo el mundo que lo lea entenderá todo el documento.

Partiendo de esta problemática el NIST (National Institute o Standards and Technology)[2] elabora un grupo de estándares que conforman SCAP[3] (Security Content Automation Protocol) con el objetivo de unificar y homogeneizar los diferentes sistemas de definición e identificación de información referente a la gestión de seguridad. Esto es

• E-mail de contacto: santiagosigo@gmail.com
• Mención realizada: Tecnologies de la Informació
• Trabajo tutoritzado por: Ángel Elbaz (DEIC)
• Curso 2020/21

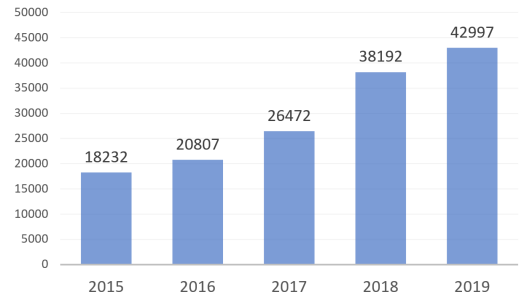

Fig. 2: Número de incidentes gestionados por el CCN-CERT entre 2015 y 2019.[1]

establecer un formato común en el ámbito de la gestión de la seguridad de los sistemas de forma que sea fácil la documentación y sea comprensible entre diferentes sistemas.

2 Objetivos del proyecto

Este trabajo se centrará primeramente en presentar SCAP a nivel teórico presentando su contexto, finalidad y que componentes presenta. También una revisión sobre herramientas open source y profesionales para implementar o que implementan el protocolo.

A partir de ahí se desarrollará un pequeño entorno de trabajo simulando un entorno cercano a la realidad para probar la aplicación de este protocolo mediante la herramienta open source OpenSCAP[4].

La metodología en la parte teórica ha sido desglosar el contenido respecto a SCAP e ir abordando los fragmentos. Ya que SCAP es como una colección de componentes es fácil dividir la información por componentes e ir uno por uno. Para la parte práctica una vez preparado el entorno, me he ido centrando en hacer los tests y analizarlos máquina por máquina.

3 Qué es SCAP

En 2007 nace el ISAP (Information Security Automation Protocol) y junto con el primer concepto de SCAP. Esta iniciativa estaba impulsada por las agencias NIST, la OSD (Operational Services Division), el DHS (Department of homeland Security), la NSA (National Security Agency) y la DISA (Defense Information Systems Agency).

En Julio de 2010 se publica la primera versión de SCAP, la versión 1.0. Meses después de revisiones se publica la versión 1.2. Y finalmente en septiembre de 2011 se publica la versión 1.3 y es la última hasta la fecha.

SCAP es una especificación abierta sobre estándares, lenguajes y convenios, conocidos como componentes, para homogeneizar y así poder automatizar la seguridad de los sistemas. SCAP está principalmente promovida por el NIST pero no está sujeto a ningún tipo de copyright, por lo que es accesible para todos y existe la libertad de adoptarlo con o sin modificaciones.

Los componentes de SCAP se pueden clasificar en varias categorías según su finalidad dentro del protocolo. Existen los siguientes tipos y los componentes que forman parte de SCAP:

- Enumeraciones
  - CVE: Common Vulnerability and Exposures
  - CCE: Common Configuration Enumeration
  - CPE: Common Platform Enumeration
- Mediciones
  - CVSS: Common Vulnerability Score System
  - CCSS: Common Configuration Score System
- Lenguajes
  - XCCDF: Extensible Configuration Checklist Description Format
  - OVAL: Open Vulnerability and Assessment Language
  - OCIL: Open Checklist Interactive Language
- Formato de informes
  - ARF: Asset Reporting format
  - AI: Asset Identification
- Integridad
  - TMSAD: Trust Model for Security Automation Data
  - SWID: Software Identification

4 Componentes de SCAP

4.1 CVE

El Common Vulnerability and Exposures[5] es un sistema de nominación y documentación estandarizado o unificado para vulnerabilidades y fallos de seguridad en un sistema o infraestructura.

Los CVEs suelen tardar días o meses en publicarse debido que se puede acordar un embargo temporal con el objetivo de solventar dicha vulnerabilidad. Una publicación directa puede suponer un riesgo pues no existe una solución conocida en el momento de la publicación.

4.2 CCE

El Common Configuration Enumeration[6] es otro componente importante en las bases de SCAP. Su objetivo es la elaboración de un sistema para la identificación de las configuraciones de los sistemas relacionadas con la seguridad de los sistemas. Es un sistema muy similar al anterior explicado CVE, pero referido a configuraciones en lugar de vulnerabilidades.

Su estructura y funcionamiento sigue una estructura similar que consiste en un identificador único que identifica dicha configuración, una descripción simple de la configuración, los parámetros en cuestión que definen la enumeración, las posibles soluciones que puede haber para el problema relacionado con la configuración en cuestión y unas referencias a los datos o herramientas en detalle.

4.3 CPE

El Common Platform Enumeration[7] es el componente de SCAP que comparte similitudes en estructura y objetivo con el CVE y CCE. De la misma forma que los anteriores se centran en identificar las vulnerabilidades y las configuraciones de los sistemas, el CPE es un sistema de identificación de tipos de aplicaciones, sistemas de operativos y dispositivos hardware dentro de los sistemas.

Esta información no refiere a instancias concretas, sino que identifica componentes de forma abstracta, indica el tipo de dispositivo que es.

Este sistema permite que ciertas herramientas puedan automatizar las decisiones de la gestión de la seguridad porque pueden acceder al CPE de los dispositivos, aplicaciones o sistemas operativos para en función del tipo de elemento actuar acorde. Por ejemplo, sabiendo que el sistema dispone de un sistema Linux kernel 3.4.1 se pueden hacer búsquedas de posibles vulnerabilidades relacionadas con dicho sistema operativo en esa versión.

4.4 CVSS

Entrando en los componentes que se consideran métricas tenemos primero el Common Vulnerability Score System[8]. Este sistema es, como su nombre en inglés indica, un sistema de puntuación de las vulnerabilidades en función de sus características a la hora de determinar la gravedad de la misma.

Esta puntuación se divide en tres principales grupos: la puntuación base, la puntuación temporal y la puntuación ambiental.

La puntuación base solo valora elementos propios o intrínsecos a la propia vulnerabilidad. Esta puntuación valora del 0 al 10 y tiene en cuenta la dificultad de explota, el alcance de la misma o los impactos en confidencialidad, integridad y disponibilidad entre otros aspectos. Esta puntuación se puede expresar como una cadena o vector donde están codificados todos los aspectos que se valoran en este sistema. Esto hace que dada una vulnerabilidad se puedan identificar los aspectos más importantes usando solo el vector.

La valoración temporal modifica partiendo de la puntuación base teniendo en cuenta valores temporales que pueden darse como la disponibilidad de parche de seguridad que lo solucione.

Y por último está la puntuación ambiental, que al igual de la puntuación temporal parte de la puntuación base para determinar si existen agravantes o atenuantes en las condiciones necesarias para poder explotar la vulnerabilidad.

4.5 CCSS

De la misma forma que el CCE es un derivado del CVE para utilizar un sistema de identificación común, el Common Configuration Score System[9] deriva del CVSS para aportar una métrica para medir la gravedad de los fallos de seguridad abiertos por parte de configuraciones dentro de los sistemas.

Al igual que el CVSS su estructura permite sintetizar la información más relevante de las configuraciones y sus posibles fallos.

4.6 XCCDF

Una vez tenemos la información clasificada y evaluada conviene estandarizar el formato en el que se almacena y organiza. Para esto SCAP incluye algunos componentes que se muestran a continuación.

El primero es el Extensible Configuration Checklist Description Format[10]. Este formato es una especificación de XML para dar un formato que sea útil a la hora de generar reportes o intercambiar información.

Las security checklists, como se puede intuir, son unas listas de elementos de la seguridad que deben revisar y cumplir en un sistema o software. El XCCDF presenta un uso estandarizado en XML para estas listas permitiendo que su fácil exportación y comunicación.

4.7 OVAL

El Open Vulnerability and Assessment Languaje[11] consiste en un lenguaje abierto y libre de uso para la estandarización de la evaluación y reporte del estado de las máquinas o sistemas informáticos.

Los tres principales pasos que sigue OVAL son: representar la información, expresar los estados específicos de la máquina y reportar los resultados de la evaluación. Todo esto se hace de forma precisa, consistente y de forma que se pueda automatizar la toma de decisiones.

4.8 OCIL

El Open Checklist Interactive Languaje[12] define un marco referencial con el objetivo de abordar esa serie de checklists menos automatizables y que requieren abordarlos de forma más manual, pudiendo complementar otros componentes como son OVAL o XCCDF.

OCIL define un formato estándar de rellenado de formularios para ser consistente y poder elaborar listas de requisitos, o checklists, de forma fácil y consistente

4.9 ARF

El siguiente grupo de componentes se centra en la creación de formatos orientados a la creación de los reportes y la documentación.

El Asset Reporting Format[13] es el componente cuyo objetivo es la de aportar un formato que permita reportar los activos de forma eficiente agilizando la relación entre todos esos datos de los anteriores componentes con el formato final del reporte de la seguridad.

Como otros componentes, el uso de una especificación común entre organizaciones y sistemas es capaz de agilizar la comunicación y automatización de la seguridad de los sistemas.

4.10 AID

El Asset Identification[14] es una especificación capaz de aportar los mecanismos con los que identificar los activos de una organización.

Esto es de gran ayuda a la hora de relacionar informaciones del sistema con los activos, lo que es un mecanismo muy útil a la hora de extraer información sobre la seguridad necesaria.

4.11 TMSAD

El Trust Model for Security Automation Data[15] presenta una serie de recomendaciones orientadas a mejorar el cómo se documenta cierta información, como pueden ser firmas, hashes o información de claves; sobre la automatización de la seguridad.

Este modelo de confianza está elaborado sobre el Lenguaje de Marcado Extensible (XML) ya que este lenguaje es muy versátil.

4.12 SWID

El Software Identification[16] es un sistema de etiquetas definido por las norma ISO / IEC 19779-2: 2015 con el objetivo de poder identificar los productos de software que hay en el mercado. El Software es esencial en las organizaciones actuales, y a menudo es utilizado una gran variedad de productos de software, por eso es importante mantener un registro preciso y actualizado de que software dispone la empresa y en que versiones está.

Un inventario de los productos de software conlleva varios beneficios. Primeramente, si se se tiene una lista de todo el software es utilizado es fácil ver cuando existe redundancia o existe cierto software que no está siendo utilizado. Cuanto menor sea la superficie de software de la empresa menor es la exposición a ser vulnerable.

Y segundo, una lista de todo el software y sus versiones facilita saber que software realmente se usa, bajo que versiones y por ende bajo que vulnerabilidades. Y este control de software facilita la planificación de actualizaciones.

Desde el NIST se recomienda e incentiva a las empresas la adopción de dicho sistema de etiquetas para que las organizaciones puedan tener sus listas de versiones utilizando este sistema.

5 Estado del arte

Una vez explicado que es SCAP y que aportan cada uno de sus componentes a esta estandarización de la seguridad en las infraestructuras de red, veremos aplicaciones en el mundo real de dicho protocolo por parte de software libre y propietario.

El NIST dispone de una serie de herramientas y módulos certificados para la implementación de SCAP de sus diferentes versiones (1.0, 1.1, 1.2 y 1.3). De entre ellos he elegido algunos para comentarlos y mostrar un poco dichas herramientas utilizadas a día de hoy[17]. He elegido de entre ellos el Security Center 5 de tenable, Nexpose de Rapid7 y OpenSCAP de RedHat.

5.1 Security Center 5

Security center 5[18] es el software de gestión de seguridad propiedad de tenable[19]. También conocido como tenable.sc, es un gestor basado en la tecnología Nessus[20] también desarrollada por tenable para el escaneo y evaluación de vulnerabilidades.

Security Center es uno de los software más elaborados y completos del mercado con respecto a la gestión de la seguridad de una infraestructura. tenable.sc es presenta la capacidad de elaborar escaneos de vulnerabilidades tanto de forma pasiva y periódica tanto como de forma activa. Tiene una cobertura de más de 56.000 vulnerabilidades debido a su integración con CVE y CCE.

Security Center 5 implementa además métricas de evaluación de las vulnerabilidades y fallos pudiendo dar información de forma intuitiva sobre los riesgos que corren los componentes de la infraestructura.

Tenable.sc además de implementar estas funcionalidades presenta una interfaz intuitiva y altamente personalizable para presentar todos esos datos que es capaz de obtener y procesar sobre el estado del sistema. Ver el apéndice A.1 para ver un dashboard de ejemplo de Security Center.

5.2 Nexpose 6

Nexpose 6[21] es la herramienta que provee Rapid7[22] para el análisis de vulnerabilidades y riesgos dentro de una infraestructura de red.

Al igual que Security Center 5, Nexpose permite la creación de múltiples tipos de escaneos de vulnerabilidades o riesgos y su automatización para que sea más fácil mantener actualizada la información del sistema.

5.3 OpenSCAP Workbench

OpenSCAP[4] es la opción open source de entre los 3 que he elegido. OpenSCAP es una herramienta más sencilla que las anteriores ya que no dispone de sistema de aprendizaje automático ni otras medidas más potentes. Pero en lo referente a lo funcional es más que suficiente.

Usando OpenSCAP Base[23] se accede a las funcionalidades de manejar escaneos usando checklists mediante línea de comandos. También existe OpenSCAP workbench[24] que es una interfaz gráfica que permite ser más intuitiva. Esta interfaz es algo más austera que las anteriores vistas.

OpenSCAP presenta funcionalidades complementarias a OpenSCAP Base y Workbench, como SCAP Daemon[25] para mantener en segundo plano evaluaciones continuadas de los sistemas.

SCAPtimony es una herramienta que sirve para centralizar los resultados de los escaneos en caso de querer tenerlos almacenados con un acceso centralizado y sencillo.

6 Comparación de herramientas y elección

Por desgracia, OpenSCAP es la única herramienta de SCAP open source que he podido encontrar, lo que hace algo injusto comparar las capacidades contra herramientas profesionales comerciales. Pese a ello, haré una valoración a modo de resumen sobre las ventajas de cada una de ellas.

Security Center 5 y Nexzpose 6 son muy parecidas en cuanto a capacidades y funciones. Son excelentes herramientas y tienen una gran escalabilidad para funcionar en grandes infraestructuras como la de una gran empresa.

OpenSCAP es una opción más modesta que como ventaja tiene su accesibilidad puesto que es de acceso libre y dispone de lo necesario para funcionar múltiples sistemas operativos. Es una herramienta más sencilla y es intuitiva de usar.

7 Entorno de trabajo

Para llevar a cabo la parte más práctica de este proyecto he dispuesto un entorno sencillo de prueba que me permita testar algunas checklists con OpenSCAP para poder probar experimentalmente las funcionalidades del protocolo SCAP

y la herramienta OpenSCAP. A continuación se describirá como he diseñado y construido el entorno de trabajo.

7.1 Diseño del entorno

Para la parte práctica del proyecto he decidido hacer una simulación simple de una infraestructura de red dentro de las posibilidades del hardware que dispongo. La infraestructura pertenece a una pequeña gestoría de contabilidad donde trabaja un jefe y 2 empleados. Por ende, los 3 dispositivos están dentro de la misma red local, utilizando sistemas operativos Linux. He decidido usar las distros Ubuntu, Debian y Fedora para las máquinas

Dentro del entorno el jefe utiliza la máquina Ubuntu y utiliza principalmente el cliente de correo Gmail a través de su navegador Firefox. Los dos empleados utilizan las máquinas Fedora y Debian. Los empleados trabajan con licencias de Cloud, por lo que no tiene software instalado y acceden desde sus respectivos navegadores, también Firefox en nuestro caso.

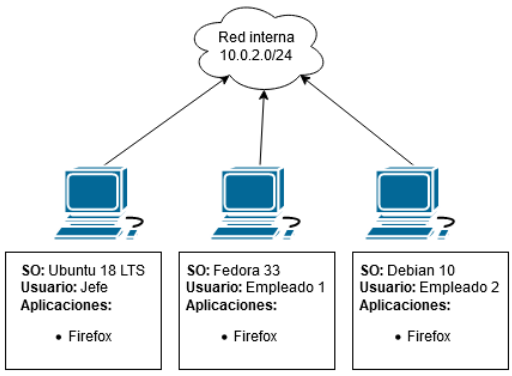

Fig. 3: Diagrama sobre el entorno de red de la simulación.

7.2 Construcción del entorno

Para la construcción de las máquinas he usado el software de virtualización libre VirtualBox [2] de Oracle. Todas máquinas han sido creadas con 2GB de RAM y se han usado las respectivas ISOs de instalación oficiales para instalar los sistemas operativos con sus propios procesos de instalación.

He decidido conectarlas entre ellas mediante una red NAT virtual a través de la opción propia de VirtualBox, como si estuvieran en la misma red local. La red es la 10.0.2.0/24 y dispone de asignación de IPs mediante DHCP. Por defecto la configuración de Virtualbox impide a las máquinas comunicarse entre ellas.

A partir de ahí lo siguiente es proceder con la instalación de OpenSCAP, aunque previamente se deben actualizar los repositorios. A continuación, se explica cómo se ha hecho en cada uno de los sistemas operativos que componen la práctica.

7.2.1 Instalación en Ubuntu/Debian

Para para instalar OpenSCAP Base hemos empezado actualizando los paquetes y utilizando el comando:

```
$ sudo apt-get install libopenscap8
```

Para la instalación de la interfaz gráfica, OpenSCAP Workbench, es necesario el siguiente comando:

```
$ sudo apt-get install openscap-workbench
```

7.2.2 Instalación en Fedora

Para instalar OpenSCAP Base en Fedora es necesario actualizar los paquetes y ejecutar el comando:

```
$ sudo dnf install openscap-scanner
```

Para la instalación de la interfaz gráfica, es necesario el siguiente comando:

```
$ sudo dnf install openscap-workbench
```

8 Aplicación de políticas de seguridad

Una vez instalado OpenSCAP Base y OpenSCAP Workbench podemos proceder con la aplicación de políticas de seguridad para cada una de las máquinas. Para empezar, hemos recurrido principalmente a SCAP seciurity Guide[27], que es un paquete, disponible para la mayoría de distribuciones, que contiene una serie de checklists básicas para evaluar dichos sistemas. El paquete se puede instalar en Fedora con el siguiente comando:

```
$ sudo dnf install scap-security-guide
```

Y en los sistemas Debian y Ubuntu mediante el siguiente:

```
$ sudo apt install ssg-base ssg-debderived
ssg-debian ssg-nondebian ssg-applications
```

8.1 Aplicación en la máquina Ubuntu 18 LTS

A raíz de este caso he decidido aplicar un escáner general al sistema operativo para detectar vulnerabilidades propias del sistema, para evitar que los datos personales del usuario se puedan ver comprometidos, y un escáner de la aplicación Firefox para evitar que sus comunicaciones o incluso sistema se puedan ver afectados.

Dentro de la checklist ssg-ubuntu1804-ds.xml he escogido el perfil intermedio de evaluación (Profile for ANSSI DAT-NT28 Average Level). Este perfil incluye una serie de tests sobre privilegios de usuario, comprobación de las particiones y tests sobre activación, desactivación y desinstalación de ciertos servicios del sistema.

Este conjunto de pruebas otorga una visión general de los aspectos más simples sobre el sistema. Después de la ejecución, se puede ver que el test ha superado el 56,94% de las reglas. De entre las reglas no superadas hay 1 que supone un riesgo alto, 6 que presentan un riesgo medio y otras 4 que presentan un rango bajo.

Se observa que 5 de ellas, las 4 leves y 1 intermedia, corresponden al particionado del disco, indicando que ciertos paths como /home, /var, /tmp, /var/log y /var/log/audit no se encuentran en particiones separadas. Esto es debido a que la instalación se ha seguido según la configuración sugerida.

Otras 3 reglas fallidas de gravedad media pertenecen a los logs, donde el sistema no rota los logs periódicamente y no pertenecen al grupo y usuario adecuado.

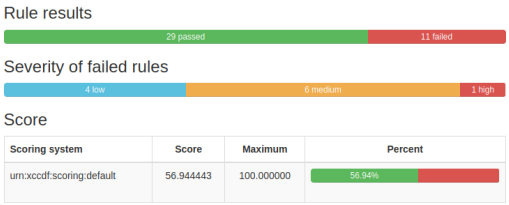

Fig. 4: Resumen de la ejecución del escáner Intermedio en Ubuntu 18

Otro aspecto fallido es la restricción de programas a patrones peligrosos. Estas reglas comprueban si las aplicaciones tendrían acceso a alterar funciones sensibles del sistema. En este apartado hay 2 fallos de gravedad media.

La regla fallida grave es la que comprueba el servicio NTP. el NTP (Network Time Protocol) permite a las máquinas conectarse a un servidor de tiempo para sincronizar periódicamente los relojes. Este servicio permite que los logs sean consistentes.

Como conclusión de este escáner, se puede ver que la mayoría de fallos no suponen un peligro, ya que el particionado, los logs y el servicio NTP no suponen un riesgo de por sí, sino más bien una falta de adhesión a las buenas prácticas en la configuración.

8.2 Aplicación en la máquina Debian 10

Para este escáner he utilizado el archivo ssg-debian10-ds.xml utilizando el perfil Profile for ANSSI DAT-NT28 Average Level que es igual en estructura que el anterior realizado en el sistema Ubuntu.

Después de la ejecución el perfil ha devuelto una puntuación de 37,78 sobre 100. De las 45 reglas, 24 se han superado, 13 no se han superado y 8 de ellas han dado error. Estas 8 reglas son las que han provocado una nota inferior a lo que se ha podido observar por ejemplo en Ubuntu.

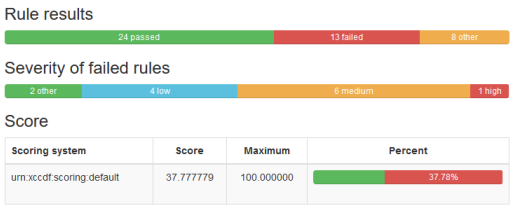

Fig. 5: Resumen de la ejecución del escáner Intermedio en Debian 10

Analizando las reglas fallidas, al igual que con la máquina Ubuntu, existe una úncia partición con todos los archivos y esto hace que 5 reglas estén fallidas en este apartado.

En el ámbito de permisos vuelven a fallar las mismas 2 reglas que en Ubuntu, pero además falla la verificación de que el archivo local System.map solo sea legible por el usuario root.

Con respecto a los logs, falla la rotación de los mismos. El perfil ha detectado que no hay ningún syslog configurado. Esta es una herramienta para gestionar los logs y poder almacenarlos en un servidor. En el aspecto de permisos y propiedad de los archivos de logs el test no ha podido identificar de quien son y ha devuelto error en las reglas.

En el servidor SSH las 5 reglas han dado error, esto es debido a que el servidor SSH no se encuentra instalado en la máquina. Estos errores se pueden obviar.

Ha fallado la regla que revisa que solo se estén utilizando uso de repositorios oficiales. Esto es debido a que en el archivo sources.list se han añadidos repositorios que no se reconocen como oficiales, estos repositorios los he añadido yo durante la configuración del sistema y preferí dejarlo para poder comentarlo. Si los repositorios no son de confianza puede conllevar problemas con software malicioso, por ejemplo.

Y finalmente, al igual que con Ubuntu, no se ha encontrado un servidor NTP configurado.

En conclusión de esta aplicación, en general es un resultado muy similar al de Ubuntu, su mala puntuación corresponde principalmente a esos tests que han dado error, dado que el servidor SSH no se encuentra instalado se puede asumir que su estado no es tan perjudicial como indica la métrica. A pesar de ello presenta fallos en reglas que Ubuntu no presentaba.

8.3 Aplicación en la máquina Fedora 33

Para la ejecución del escáner básico he usado la chacklist ssg-fedora-ds.xml y he aplicado el perfil Standard System Security Profile for Fedora. Este test difiere en estructura de los de Debian y Ubuntu, pero en contenidos es similar e incluso más completo.

Después de ejecutar el escáner ha obtenido una puntuación de 57,42 sobre 100. De este escáner de 79 reglas, se han superado 30 y no se han superado 49. De entre esas 49, según openSCAP 2 de ellas son de elevada importancia, 45 media y otras 2 leve o nula.

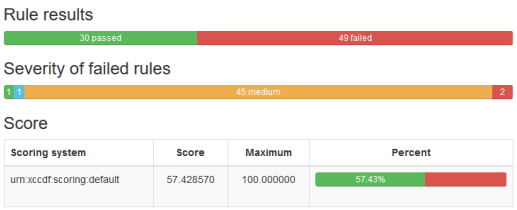

Fig. 6: Resumen de la ejecución del escáner Intermedio en Fedora 33

Entrando en el detalle de que tests no se han superado, en el ámbito del usuario del sistema se ha detectado que no se notifica la última conexión, no se establecen máximos de edad y longitud de contraseña, no se previene de hacer login con el campo de contraseña vacío y el acceso directo a root está activo.

El grueso de los tests fallidos se encuentran en el apartado referente a auditd. Esta sección incluye múltiples tests sobre si se registran ciertos eventos o sucesos. El Sistema por defecto no está configurado y no está guardando la mayoría de los datos que revisa el perfil. Esta herramienta es útil no tanto para prevenir sino para entender o ver que está sucediendo, como los intentos de acceso no autorizados por ejemplo.

En lo referente a medidas de la integridad del software existen 3 reglas no superadas. Ha fallado el test de la base de datos AIDE, la verificación de permisos de archivos

del RPM y la configuración de OpenSSL para poder usar Crypto Policy.

En términos de privilegios hay 2 reglas fallidas, la primera es que el soporte de kernel para el bootloader en USB esta activado y no debería estarlo. La segunda es que no todos los archivos ejecutables del sistema pertenecen al usuario root.

Y en el ámbito de las redes, se ha detectado que el parámetro Defaultzone no se encuentra puesto en drop.

En conclusión del perfil, este test es más riguroso que los demás. En términos de reglas fallidas, no existen errores graves directos, más bien falta de adhesión a buenas prácticas o aspectos que facilitan ataques. Pero en general el sistema es bastante seguro, aunque mejorable.

8.4 Aplicación en el navegador Firefox

Todas las máquinas utilizan Firefox como herramienta de trabajo y parten de la configuración predeterminada del mismo. Por eso en lugar de aplicar 3 tests he decidido simplificarlo en un único test ya que la configuración de la aplicación es independiente del sistema operativo.

Para este escáner se ha utilizado la checklist ssg-firefox-ds.xml que contiene un único perfil, Upstream Firefox STIG, que esta compuesto de 22 reglas.

Después de ejecutarlo, se puede ver que es el que peor puntuación presenta con un 5,26 sobre 100. Este test solo ha superado 1 regla, referente a la versión del navegador.

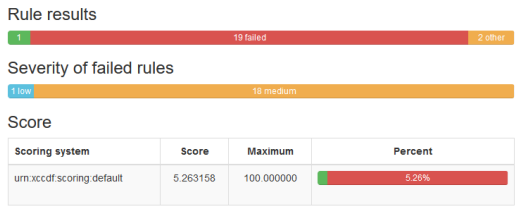

Fig. 7: Resumen de la ejecución del escáner en Firefox

De entre todas las pruebas no superadas hay 2 que no se ha podido revisar, son referentes a los certificados. No se ha encontrado el certificado DoD root y por ende no se encuentra compartido con el sistema. Este certificado es certificado del Department of Defense, el equivalente estadounidense del certificado digital de persona física en España.

El resto de pruebas pertenece a parámetros del navegador. La lista de completa de reglas fallidas es la siguiente:

- Instalación de extensiones activada.
- Descarga automática de contenido MIME activada.
- Uso de TLS desactivado.
- Capacidad de JavaScript de mover o redimensionar ventanas activada.
- Menús contextuales de JavaScript activados.
- Confirmación de descarga o apertura de archivos desactivada.
- Asistente de auto-rellenado de formularios activada.
- Capacidad de auto-rellenado de contraseñas de usuario activada.
- Herramientas de desarrollo de Firefox activadas.
- Bloqueador de Pop-up de Firefox desactivado.
- Acceso de Firefox a protocolos del shell activado.
- Verificación de certificados desactivada.
- Envío de información de uso en segundo plano activada.
- Comprobación de actualización de los plugins de búsqueda activada.
- Actualización automática de las extensiones activada.
- Almacenamiento de contraseñas de Firefox activado.
- Capacidad de JavaScript de mover delante o atrás las ventanas activada.

En conclusión, Firefox de forma predeterminada no implementa la mayoría de medidas de seguridad que comprueba la checklist. La mayoría de opciones no suponen peligros directos o potenciales. Ya que por ejemplo las referentes a JavaScript no son peligrosas, pero pueden ser usadas por atacantes para enmascarar ataques. También que es cierto que hay opciones como el uso de certificados y TLS que son importantes.

9 Conclusiones de la aplicación de las políticas

Después de la aplicación de todos los perfiles se puede observar que en general todos tienen fallos que pueden ser remediados para alcanzar un nivel de cumplimiento con el perfil bastante mayor. En algunos casos puntuales hay medidas que son decisión de la empresa, como el caso del servidor NTP el cual no tiene por qué ser necesario. Firefox por ejemplo sí que presenta genuinamente un mal resultado en el perfil aplicado y aunque la gran mayoría de aspectos no suponen riesgos directos suponen dar facilidades, información sensible y herramientas a los atacantes que hagan un ataque usando el navegador como punto de entrada.

10 Resolución de los problemas encontrados

Una vez ejecutados los tests para estos sistemas y aplicación explicaré a continuación los aspectos más importantes a remediar de cada uno.

Los reportes de OpenSCAP aportan información resumida sobre todo los tests, impacto en el sistema, posible solución y en algunos casos la posibilidad de generar un script de remediación para el problema en cuestión.

10.1 Resolución de los problemas en Ubuntu

Después de observar los resultados del análisis intermedio de Ubuntu se puede ver que en general, la instalación por defecto de Ubuntu no presenta errores especialmente cruciales para el uso promedio en una oficina.

Pese a ello hay aspectos mejorables que comentaré a continuación.

10.1.1 Particionado

Empezando por el particionado, se observan errores de que las carpetas /home, /var, /tmp, /var/log y /var/log/audit no estén en particiones separadas. Esto es debido a que durante la instalación se ha seguido el proceso predeterminado de particionado que hace una instalación en una única partición.

No es necesario que cada directorio tenga una partición exclusiva, sería recomendable separarlas temáticamente.

Mantener estas carpetas en particiones externas permite un mejor control de acceso y montaje de las mismas y permite controlar el espacio usado para que solo contenga los archivos necesarios.

Este aspecto se puede solucionar en la propia instalación eligiendo crear particiones para algunos de estos directorios, o se pueden crear las particiones y migrar los directorios, que en este caso seria el método a seguir.

10.1.2 Sistama de logs

En al apartado de los logs existen 2 tipos de alertas. Dos de ellas hacen referencia a la propiedad de los archivos de log del sistema y una última referente a la rotación de los logs. Los logs deben tener como propietario al usuario root, esto puede incurrir en un problema ya que los logs pueden contener información sensible y deben estar protegidos de accesos no autorizados.

Para este caso, OpenSCAP nos da la solución indicando los siguientes comandos para cambiar el usuario y grupo propietario respectivamente:

```
$ sudo chown adm LOGFILE
$ sudo chgrp adm LOGFILE
```

El otro aspecto no superado es la rotación de los logs del sistema, esta opción impide que el archivo de logs crezca sin control con el paso del tiempo. OpenSCAP recomienda ponerlo en diario. Para ello se puede añadir la siguiente línea en /etc/logrotate.conf:

```
rotate log files frequency daily
```

10.1.3 Permisos de aplicacones

El primer error relacionado con los privilegios indica que los core dumps están activados. Los core dumps son archivos que pueden guardar información de la configuración o el estado de un proceso en caso de que este fallara. Esta opción OpenSCAP recomienda deshabilitarla debido a que puede almacenar información sensible. Para modificar de forma persistente esta opción se debe añadir la siguiente línea al archivo /etc/sysctl.d:

```
fs.suid_dumpable = 0
```

El segundo aspecto que menciona el test es que la aleatorización del espacio de direcciones esta desactivada. Aleatorizar el espacio de direcciones dificulta a los atacantes predecir la localización del código malicioso que han introducido en el espacio de direcciones de un proceso durante el ataque. Esta opción se puede modificar añadiendo al siguiente línea en el archivo /etc/sysctl.d:

```
kernel.randomize_va_space = 2
```

10.1.4 Instalación del NTP

El servicio NTP es recomendable para mantener consistencia en la hora de los dispositivos. Puesto que en nuestra oficina solo se disponen de 3 dispositivos y todos tienen acceso a internet no sería necesario pues con la propia configuración de fecha y hora mediante el acceso a internet sería suficiente.

10.2 Resolución de los problemas en Debian

En Debian debido a que la checklist es muy similar y que tanto Debian como Ubuntu son internamente muy similares los resultados son bastante parecidos, y por ende, sus soluciones también. No obstante, hay diferencias con el anterior comentado. A continuación, se abordan todos los aspectos relevantes del resultado.

10.2.1 Particionado

En este apartado tenemos el mismo caso que en Ubuntu, al seguir la instalación predeterminada del sistema los directorios /tmp, /home, /var, /var/log y /var/log/audit no se encuentran en particiones separadas. Como ya se ha comentado se pueden crear particiones y migrar los directorios existentes.

10.2.2 Permisos de aplicaciones

En este apartado también se dan las 2 alertas que en Ubuntu se presentaban y se solucionan de la misma forma que en Ubuntu.

Además, se observa una alerta referente a los permisos de acceso al archivo System.map. Este archivo contiene información sensible del sistema y debería tener acceso restringido. Usando el siguiente comando que sugiere OpenSCAP se puede solucionar:

```
$ sudo chmod 0600 /boot/System.map-*
```

10.2.3 Sistemas de logs

En el sistema de logs se observa que el servicio rsyslog ha devuelto Unkown, esto puede ser debido a que el servicio rsyslog no está instalado. Es necesario tenerlo instalado puesto que el daemon que se encarga de los logs del sistema. Para instalarlo, si no lo estuviera, solo hace falta el siguiente comando:

```
$ apt-get install rsyslog
```

Adicionalmente los tests sobre propiedad y permisos han dado Unkown. Aquí el test ha devuelto algún error y no ha podido determinar el estado. De todas formas, se pueden aplicar los comandos mencionados para Ubuntu para asegurarse.

También se ha observado que la rotación de los logs esta desactivada, como en Ubuntu. Usando el mismo comando se puede habilitar con el periodo deseado.

10.2.4 Configuración de servidor SSH

Todos los tests con respecto al servidor SSH han dado Unkwown. Esto es debido a que el servidor SSH no se ha instalado durante la instalación del sistema operativo.

Debido a que es un ordenador personal para el trabajo no debería ser necesario instalar un servidor SSH en esta máquina. Si fuera necesario se podría instalar en el futuro.

10.2.5 Configuración sel servicio APT

Por defecto Debian solo incluye sus propios repositorios en souerce.list. Para arreglar esto es simplemente entrar el archivo sousrces.list y eliminar aquellos repositorios que no

sean oficiales. Si dichos repositorios fueran confiables y necesarios no haría falta eliminarlos.

10.2.6 Instalación del servicio NTP

Al igual que en Ubuntu, como la infraestructura no es lo suficientemente grande y compleja no sería necesario instalar un servidor NTP.

10.3 Resolución de los problemas en Fedora

El perfil de la checklist utilizado en Fedora es más exhaustivo y presenta más deficiencias.

10.3.1 Control de acceso

En el acceso se observa que esta deshabilitada la opción que permite indicar el último login al usuario. Esto es recomendable porque permite identificar accesos no autorizados al sistema. esto se puede arreglar añadiendo las siguientes líneas en el archivo /etc/pam.d/postlogin:

```
session [success=1 default=ignore]
    pam_succeed_if.so service !~ gdm* service !~ su* quiet
session [default=1]  pam_lastlog.so nowtmp showfailed
session optional  pam_lastlog.so silent noupdate showfailed
```

También se observa que no existen parámetros para regular la periodicidad del cambio de contraseñas ni la longitud máxima. Para remediarlo convendría que estuviera estipulada una duración mínima y máxima de las contraseñas, así como una longitud mínima de las mismas. Para arreglar esto se deben añadir las siguientes líneas al archivo /etc/login.defs:

```
PASS_MIN_DAYS 7
PASS_MAX_DAYS 90
PASS_MIN_LEN 12
```

Se observa que está habilitada la opción de login con contraseña vacía. Para remediar esto se deben eliminar todas las entradas del archivo /etc/pam.d/system-auth.

Los usuarios tienen acceso de login directo a root. Corregir esto supone poder manejar mejor el acceso a root de los usuarios, ya que estos por defecto no dispondrían de privilegios y tendrían que escalarlos mediante otro login. Para remediar esto se puede ejecutar el siguiente comando:

```
$ sudo echo > /etc/securetty
```

10.3.2 Configuración de los logs

En el apartado de los logs se observan 34 tests fallidos. De los cuales 30 corresponden a aspectos que no se están grabando en los logs. las soluciones son sencillas pero el volumen de elementos me impide comentarlos todos. En todos las solución consiste en añadir la respectiva línea en el archivo /etc/audit/rules.d o en /etc/audit/audit.rules si está configurado para usar auditctl. En el reporte indica todas las líneas necesarias para cada aspecto.

10.3.3 Integridad del software

Se observa que la base de datos AIDE (Advance Intrusion Detection Enviorenment) no se encuentra inicializada. Esta herramienta permite guardar las expresiones regulares de los archivos de configuración y permitir verificar la integridad del software. Se puede inicializar con:

```
$ sudo /usr/sbin/aide --init
```

el archivo generado por defecto está en /var/lib/aide/aide.db.new.gz. con ese archivo podemos renombrarlo para eliminar el new de las extensiones y usar la aplicación de aide con el comando –check para verificar con los siguientes comandos:

```
$ sudo cp /var/lib/aide/aide.db.new.gz
            /var/lib/aide/aide.db.gz
$ sudo /usr/sbin/aide --check
```

Otro problema es la verificación de permisos por parte del instalador de paquetes RPM. Permisos sobre archivos de configuración o sobre binarios del sistema demasiado generosos puede provocar que ciertos usuarios tengan acceso a modificar archivos que no deberían. Para localizar que archivos no cumplen con los permisos establecidos por el vendedor se puede ejecutar el siguiente comando:

```
    $ sudo rpm -Va | awk '{
            if (substr($0,2,1)=="M") print $NF}'
```

Y a partir de ahí se puede localizar cada paquete y resetear los permisos con los respectivos comandos:

```
$ rpm -qf [Nombre del archivo]
$ sudo rpm --setperms [Nombre del paquete]
```

10.3.4 Políticas criptográficas del sistema

Se observa que la librería System Crypto Policy de OpenSSL no se encuentra configurada. Esto puede provocar errores en la ejecución de Java y produce mayor fragmentación de la configuración del sistema. esto se puede remediar añadiendo la siguiente línea en el archivo /etc/pki/tls/openssl.cnf:

```
.include /etc/crypto-policies/back-ends/opensslcnf.config
```

10.3.5 Permisos de archivos

OpenSCAP indica que la configuración de bootloader vía USB se encuentra activa. Esto puede suponer un riesgo potencial contra dispositivos USBs maliciosos. Pero es cierto que no supone un riesgo elevado y puede afectar a periféricos USB provocando problema. Por eso no es necesario modificar nada en este aspecto.

El perfil exige que todos los ejecutables del sistema tenga como propietario al usuario root. Esto limita más el alcance de los usuarios a la hora de modificar aspectos del sistema. Para modificarlo debe usar el comando mencionado anteriormente chown en los ejeutables del sistema, que suelen encontrarse en los paths /bin, /sbin, /usr/bin, /usr/libexec, /usr/local/bin, /usr/local/sbin y /usr/sbin.

10.3.6 Configuración de red

Se observa que no existe un valor por defecto para el firewall del sistema. Se recomienda descartar por defecto, aunque si hubiera problemas se tendrían que hacer configuraciones más detalladas del firewall. Para esto es simplemente añadir o modificar la siguiente línea al archivo /etc/firewalld/firewalld.conf:

```
DefaultZone=drop
```

10.3.7 Servicios

En el tema de los servicios, los 3 fallos están relacionados con SSH. El primero corresponde a permitir el login al usuario root vía SSH. El segundo corresponde a no establecer un tiempo máximo de inactividad. y el último a establecer un máximo de mensajes de keep alive, como queremos configurar un timeout por intervalo, este último debe ponerse a 0.

Para hacer estos cambios se debe acceder al archivo /etc/ssh/sshd_config y modificar las siguientes lineas respectivamente:

```
PermitRootLogin no
ClientAliveInterval 300
ClientAliveCountMax 0
```

10.4 Resolución de los problemas en Firefox

Por defecto Firefox no implementa muchas de las opciones recomendables para asegurar la seguridad del sistema. Si bien no todas son necesarias, muchas de ellas son recomendables. A continuación, se clasifica y explica sobre los aspectos encontrados en el escáner. Algunas opciones se pueden encontrar en el menú de configuración del navegador, aunque todas se pueden cambiar modificando parámetros buscando about:config en la barra de búsqueda. Desde ahí se pueden buscar los parámetros necesarios y modificarlos.

10.4.1 Opciones de JavaScript

Un aspecto que no presenta un riesgo como tal, pero da control a los atacantes es la capacidad del uso de JavaScript para alterar las ventanas. Esto puede ser creando ventanas emergentes, mover, redimensionar, poner por delante o detrás de otras ventanas. Estos permisos permiten esconder ataques o permitir a ciertas páginas de spam tener control sobre más aspectos.

Para solucionar este problema conviene modificar los siguientes parámetros:

```
dom.disable_window_move_resize => true
dom.event.contextmenu.enabled => false
dom.disable_window_flip => true
```

10.4.2 Auto-rellenado y almacenamiento de credenciales

Cualquier asistente de auto-rellenado de formularios o credenciales implica el almacenamiento de esa información de forma local. El almacenamiento de credenciales puede suponer un riesgo si el atacante consigue acceder a esos archivos locales y pueda extraer información sensible.

Para solucionar este problema conviene modificar los siguientes parámetros:

```
browser.formfill.enable => false
signon.autofillForms => false
signon.rememberSignons => false
```

10.4.3 Opciones de descarga

La descarga y ejecución de contenidos esta automatizada por defecto. Las descargas se descargan automáticamente en la carpeta de descargas local del sistema. Firefox permite que los contenidos MIME se descarguen y ejecuten con el plugin correspondiente. Desde las preferencias del navegador en al apartado General existe la opción de descargar automático o preguntar siempre. Es recomendable que se pregunte siempre para evitar descargas no deseadas por parte de sitios maliciosos o dudosos. Para solucionar la ejecución automática de contenidos MIME y pregunte siempre si se quiere ejecutar se debe modificar el siguiente parámetro:

```
browser.helperApps.alwaysAsk.force => true
```

10.4.4 Opciones de seguridad

Se observa que el TLS (Transport Layer Security) no se encuentra en las versiones recomendadas Para activar el TLS conviene modificar las versiones mínimas y máximas de TLS. Los valores recomendados son los siguientes:

```
security.tls.version.min => 2
security.tls.version.max => 4
```

La verificación de certificados se encuentra desactivado. Esto es debido a que no se encuentra ningún certificado digital configurado. Esto es recomendable para identificarte en accesos como los relacionados con entidades públicas. OpenSCAP recomienda preguntar siempre por el certificado, esta opción ya se encuentra correctamente configurada.

Firefox implementa un bloqueador de pop-ups integrado y se recomienda activarlo. Las páginas con publicidad abusiva pueden generar múltiples ventanas sin consentimiento del usuario. esto puede afectar a otras páginas legitimas, pero Firefox avisa y puedes añadir excepciones. El parámetro a modificar es el siguiente:

```
dom.disable_window_open_feature.status => true
```

10.4.5 Opciones de extensiones

Existen un par de factores relacionados con las extensiones. Es recomendable deshabilitar la instalación de extensiones y la búsqueda automática de actualizaciones. Ambas decisiones pretenden evitar el uso de fuentes no fiables para las extensiones, ya que las extensiones pueden conseguir permisos muy amplios. Ambas opciones se pueden desactivar modificando los parámetros:

```
xpinstall.enabled => false
browser.search.update => false
```

10.4.6 Otras opciones

Como medidas adicionales se puede desactivar el envío de información hacia Mozilla. Esto favorece la privacidad de los datos al no enviar información constantemente a Mozilla.

También se observa que el acceso de Firefox al shell está activado. Esto permitiría a los atacantes ganar acceso al shell de comandos y tener acceso al sistema.

También es recomendable desactivar las opciones de desarrollo de Firefox. Estas opciones a menudo establecen logs de forma local y pueden contener información sensible.

Estas 3 opciones se pueden desactivar con los siguientes parámetros respectivamente:

```
datareporting.policy.dataSubmissionEnabled => false
network.protocol-handler.external.shell => false
devtools.policy.disabled => true
```

11 Conclusiones

En este trabajo se ha desarrollado sobre el protocolo SCAP y todos los estándares de los que se compone para ofrecer una forma sistemática de describir la seguridad de los sistemas. A partir de ahí se ha diseñado un entorno sobre el que probar la herramienta open source OpenSCAP. A raíz de una hipotética oficina pequeña con 3 máquinas de trabajo formando una infraestructura sencilla se han elaborado pruebas para ver si los sistemas podrían presentar problemas y se ha visto que en todos hay aspectos mejorables y se ha profundizado un sobre la solución de dichos problemas.

Como objetivo futuro me gustaría abarcar una infraestructura algo más compleja para poder ver más rango de aplicaciones del protocolo no solo en sistemas y en una aplicación como Firefox. También me gustaría poder trabajar con un software propietario como Nexpose o Security Center para poder comparar y para tener una experiencia más cercana a una aplicación real.

Agradecimientos

Primeramente, me gustaría agradecer a mi tutor del trabajo Ángel Elbaz por su ayuda, ya que las veces que he tenido que hablar con él ha sabido ser muy claro para poder enfocar y guiar el trabajo. También me gustaría agradecer a Óscar Mascuñano por su apoyo durante estos años de carrera. Y gracias a Alexandre Carrillo por apoyarme también en este trabajo.

Referències

[1] https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/5377-ccn-cert-ia-13-20-ciberamenazas-y-tendencias-edicion-2020/file.html
[2] https://www.nist.gov/topics/cybersecurity
[3] https://csrc.nist.gov/projects/security-content-automation-protocol
[4] https://www.open-scap.org/features/standards/
[5] https://cve.mitre.org/about/
[6] https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/Specifications/Common-Configuration-Enumeration-(CCE)
[7] https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/Specifications/cpe
[8] https://nvd.nist.gov/vuln-metrics/cvss
[9] https://www.nist.gov/publications/common-configuration-scoring-system-ccss-metrics-software-security-configuration
[10] https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/Specifications/xccdf
[11] https://oval.mitre.org/
[12] https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/Specifications/ocil
[13] https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/Specifications/arf
[14] https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/Specifications/aid
[15] https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/Specifications/tmsad
[16] https://csrc.nist.gov/projects/Software-Identification-SWID
[17] https://csrc.nist.gov/projects/scap-validation-program/validated-products-and-modules
[18] https://es-la.tenable.com/products/tenable-sc
[19] https://es-la.tenable.com/
[20] https://es-la.tenable.com/products/nessus
[21] https://www.rapid7.com/products/nexpose/
[22] https://www.rapid7.com/
[23] https://www.open-scap.org/tools/openscap-base/
[24] https://www.open-scap.org/tools/scap-workbench/
[25] https://www.open-scap.org/tools/openscap-daemon/
[26] https://www.virtualbox.org/
[27] https://www.open-scap.org/security-policies/scap-security-guide/

APÉNDICE

A.1 Dashboard de Security Center

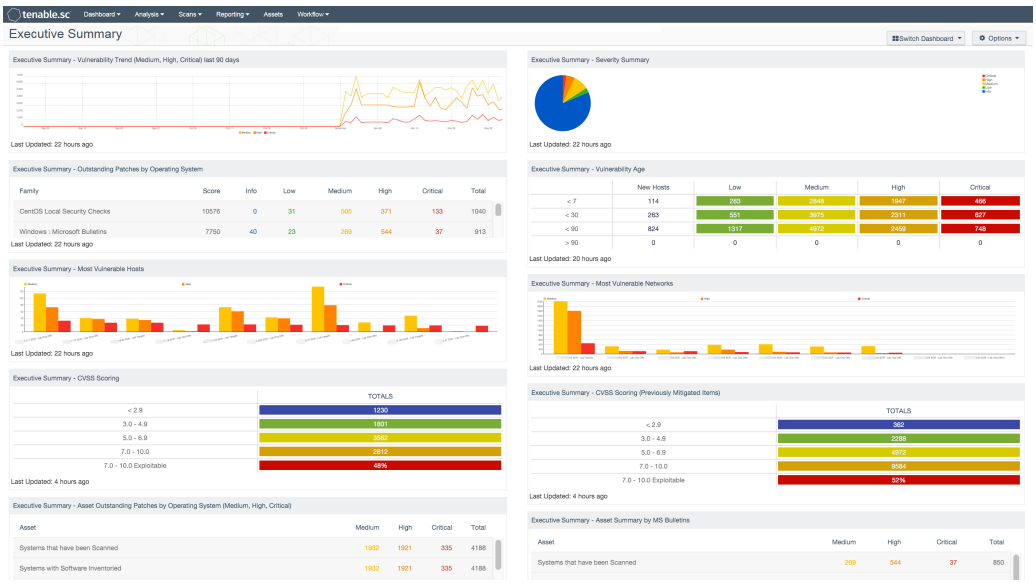

Fig. 8: Ejemplo de dashboard con Security Center 5.

A.2 Captura de la ejecución en OpenSCAP Workbench

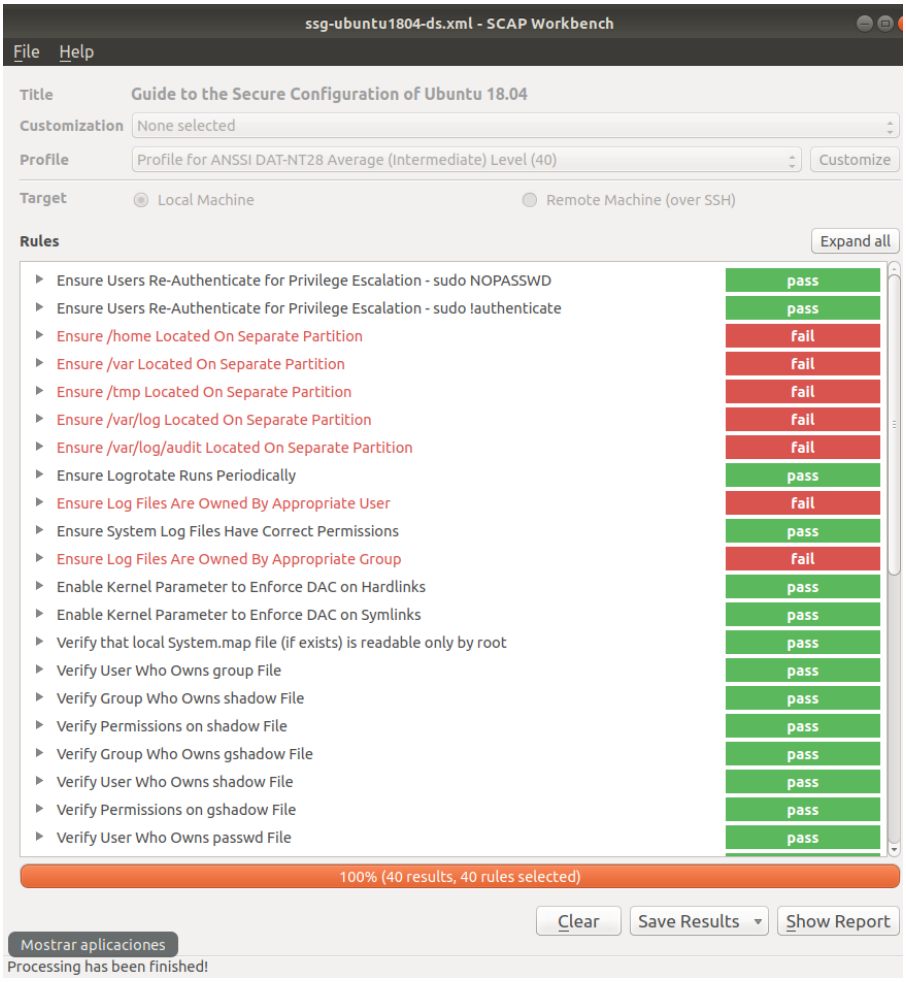

Fig. 9: Captura del perfil de Ubuntu en OpenSCAP Workbench.

A.3 Checklists utilizadas

incluidos en el archivo de entrega de este informe, en /Archivos/Checklist o usando los enlaces en el archivo index.html que llevan directamente a las respectivas checklists en formato zip.

A.4 Reportes obtenidos

incluidos en el archivo de entrega de este informe, en /Archivos/Reportes o usando los enlaces en el archivo index.html que llevan directamente a los reportes en formato html.

A.5 Imágenes

Todas las imágenes están incluidas en el directorio /Imagenes.