UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

HERRAMIENTA DE GESTION DE VULNERABILIDADES EN SISTEMAS

Sandra Cuevas López

Tutor: Laura Hernández Ardura

Ponente: Iván Cantador Gutiérrez

JULIO 2017

HERRAMIENTA DE GESTION DE VULNERABILIDADES EN SISTEMAS

AUTOR: Sandra Cuevas López

TUTOR: Laura Hernández Ardura

PONENTE: Iván Cantador Gutiérrez

Dpto. Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio de 2017

Resumen

En este proyecto se ha desarrollado una aplicación para el departamento de Seguridad de T-Systems que permite gestionar todo el ciclo de vida de la gestión de vulnerabilidades de sus clientes así como generar reportes y realizar seguimiento del estado de seguridad de los sistemas.

La aplicación tiene como entrada de datos información acerca de los servidores gestionados y scans de vulnerabilidades realizados con distintas herramientas. Una vez introducidos los datos en la aplicación, ésta los normaliza e integra para que los usuarios puedan mediante distintas tablas gestionar las vulnerabilidades. La gestión que realizan los usuarios va desde agrupación de servidores para realizar los scans de vulnerabilidades, asignación de distintos estados a las vulnerabilidades dependiendo de la fase en la que se encuentren de su ciclo y gestión de tickets que permiten que las vulnerabilidades se soluciones o se acepte su riesgo. Con todos estos datos gestionados por el usuario, la aplicación genera *dashboards* con múltiples gráficas que permiten ver se forma visual el estado general de seguridad de un cliente además de un indicador global que mide el nivel de securización de los activos. Finalmente, la aplicación permite generar reportes según la criticidad o riesgo de las vulnerabilidades en formato Word de forma automatizada, en los que se incluyen gráficas y tablas descriptivas para los clientes.

Palabras clave: gestión de vulnerabilidades, generación de reportes , seguridad, amenaza, criticidad, riesgo.

Abstract

This project is a web application for the Security Operations department of the company T-Systems. This application allows the management of the vulnerabilities from all customers and generates security reports and measure the security level of the systems.

The application has as input data information from managed servers (IP's, status, Operating system...etc.) and vulnerability scans performed with different tools. Once the data has been introduced into the application, it is normalized and integrated so that users can, through different tables, manage the vulnerabilities. The users of the application can perform multiple tasks: group servers to perform vulnerability scans, assign different status to the vulnerabilities and manage tickets that allow vulnerabilities to be resolved or accept their risk. With all this data managed by the user, the application generates dashboards with multiple graphs and allows the users to see global indications of the security of the customers. Finally, the application generates automatic reports according to the criticality or risk of the vulnerabilities in Word format, which include graphs and descriptive tables for the customers.

Keywords: vulnerability managent, generate reports, security, threat, criticality, risk.

Agradecimientos

A ti mamá, porque siempre me has dado fuerza para seguir adelante, porque mi admiración por ti, por lo que eres y por todo lo que consigues es infinita. Siempre me has apoyado en todas mis decisiones y has estado ahí siempre que lo he necesitado, como un pilar indestructible. Te quiero.

Esta carrera ha sido uno de los mayores retos a los que me he enfrentado, me ha agotado muchas veces tanto física como psicológicamente, pero no he estado sola. He tenido algunos profesores maravillosos que me apoyaron cuando no podía más, como Susana Holgado, Marina de la Cruz, David Camacho o a Iván Cantador que decidió ayudarme con este proyecto cuando parecía que encontrar un ponente era tarea imposible.

De esta etapa también me llevo a dos personas muy especiales, a Alex, a quien nunca podré agradecerle suficiente todo lo que ha hecho por mi y todo lo que me ha cuidado desde que le conocí. Y a Jimena, que apareció en mi vida de una forma inesperada y se ha ganado todo mi cariño, la quiero con locura, sé que puedo confiar y apoyarme en ella siempre que lo necesite y sé que es una amiga para siempre.

El proyecto no habría existido si no fuera por Abel, que estuvo ahí cuando entré de becaria en el departamento, me enseño casi todo lo que sé sobre seguridad, y me motivó a llevarlo a cabo, es una persona a la que admiro con todo mi corazón. Por otro lado, jamás olvidaré el día en que Laura me llevó a un despacho para hablar conmigo y me ofreció entrar en la plantilla del departamento, gracias a ella tengo una trabajo que me llena día a día y trato dar lo mejor de mi en él siempre. Nos cuida y nos defiende, con ella aprendo también todos los días cómo enfrentarme a nuevas situaciones y trato de seguir sus pasos lo mejor posible para algún día poder llegar a ser una gran profesional como lo es ella. Y cómo no, tengo que agradecer también a Darío, es un gran apoyo para mi, trabajar con él es un privilegio y siempre está ahí para hacerme reír hasta no poder más aunque estemos hasta arriba de trabajo y agotados. A todo el equipo de Security Operations, gracias, aprendo todos los días con vosotros, es un lugar genial donde trabajar.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

GLOSARIO

Nombre	Descripción
CVSS	Common Vulnerability Scoring System. Estándar en la gestión de vulnerabilidades que asigna severidad a las amenzas.
CVE	Common Vulnerabilities and Exposures. Ofrece un listado de información sobre vulnerabilidades de seguridad conocidas con un identificador único para cada una de ellas. Este listado fue definido por The MITRE Corporation.
Nessus	Herramientas de escaneo de vulnerabilidades en sistemas
OpenVAS	
Qualys	
CMDB	Configuration Management DataBase. Base de datos de activos en T-Systems. Dispone toda la información de cada uno de los activos que gestiona.
SDM	Service Delivery Manager. Figura encargada de un servicio en T-Systems. Hace de intermediario entre el cliente y la empresa.
OPM	Operations Manager. Figura que gestiona la parte operativa de un servicio. Hace de intermediario entre los equipos técnicos y la figura de SDM.
CuSM	Customer Security Manager. Figura encargada de coordinar la seguridad de un cliente en concreto.
RFXXX	El estándar seguido para la especificación del identificador de cada requisito funcional será de la siguiente manera: R = Requisito F = Funcional XXX = secuencia de tres dígitos que servirá para la enumeración de cada requisito.
RNFXXX	El estándar seguido para la especificación del identificador de cada requisito no funcional será de la siguiente manera: R = Requisito NF = No Funcional XXX = secuencia de tres dígitos que servirá para la enumeración de cada requisito.
ITIL	Information Technology Infrastructure Library. Son un conjunto de conceptos y buenas prácticas para la gestión de servicios relacionados con las tecnologías de la información. Permite medir la calidad y eficiencia de los operaciones realizas en una empresa.
SM9	Herramienta Service Manager 9 utilizada para gestión de tickets siguiendo la metodología ITIL en T-Systems
KPI	Key Performance Indicator. Indicador que se emplea para medir el nivel de securización de un servidor o conjunto de los mismos.

CAB	Change Advisory Board. Comisión encargada de revisar todos los cambios que se realizan a través de la herramienta SM9 en T-Systems. Comprueba criterios de calidad así como posibles solapamientos con otras intervenciones.
Security Operations	Departamento de Seguridad en la empresa T-Systems para el que se ha desarrollado la herramienta de gestión de vulnerabilidades presentada en este proyecto.
CERT	Computer Emergency Response Team. Equipo que da respuesta frente a emergencias relacionadas con la seguridad de la información y provee información diaria de nuevas vulnerabilidades.

Tabla 1. Glosario de términos y acónimos

1 INTRODUCCIÓN

1.1 Motivación

En los últimos años el mundo relacionado con la tecnología ha evolucionado drásticamente. Desde la llegada de internet ha cambiado la forma en que nos comunicamos, la forma en que vivimos y cómo interactuamos con las personas y el mundo que nos rodea. Hoy en día nos es posible a través de un dispositivo conectado a internet, como puede ser un Smartphone, hacer la compra, transferir dinero, montar un negocio o incluso buscar pareja. El abanico de posibilidades que tenemos al alcance de la mano es prácticamente infinito. Con la llegada de este cambio que afecta a todos los ámbitos en nuestras vidas, nos hacemos más dependientes de la tecnología aumentando los riesgos que implica depender de ella [3].

La tecnología evoluciona día a día y la mayoría de las veces no somos conscientes de lo vulnerables que podemos llegar a ser en caso de que la ésta falle. Los problemas que nos pueden afectar van desde la privacidad o seguridad de las personas hasta grandes impactos financieros para los negocios o economía. Últimamente no es difícil encontrar en las noticias casos de filtrado de información como puede ser Wikileaks, hackeo de sistemas de tráfico, coches o aviones ataques DDoS imposibilitando los servicios de una empresa e incluso cifrado de información como hospitales solicitando rescates. Todos estos casos ponen de manifiesto que el entorno en el que nos encontramos no es tan seguro como podamos creer o que desconozcamos en qué medida estamos expuestos y el impacto que puede llegar a tener. Debemos ser conscientes de los riesgos existentes para poder anticiparnos a los mismos con la finalidad de prevenir los ataques [2].

Por tanto, la ciberseguridad comienza a ser un elemento esencial para proteger los sistemas de información, ya que no solo puede tener impactos económicos en empresas sino que nos puede afectar drásticamente en nuestro día a día. Por ello debemos estar preparados, disponer de las herramientas necesarias para hacer frente a las amenazas que cada vez son más complejas y abordar los retos que se nos plantean con esta evolución [3].

1.2 Objetivos

Durante los últimos años el departamento de Seguridad de T-Systems ha visto como la solicitud de servicios de Seguridad por parte de los clientes ha ido creciendo exponencialmente, especialmente la gestión continua de vulnerabilidades en los sistemas. Este incremento ha llevado al departamento a mejorar su porfolio de servicios con el objetivo de posicionarse a la altura de sus competidores mejorando en eficiencia.

Para la mejora del servicio de gestión de vulnerabilidades, se ha decidido invertir en el desarrollo de una aplicación en la que se pueda realizar seguimiento de todo el ciclo continuo de gestión de vulnerabilidades encontradas en los sistemas, desde que se decide programar un escaneo de vulnerabilidades hasta que se da por resuelta y se reporta al cliente.

Los objetivos que persiguen el desarrollo de esta aplicación:

1. **Unificar el resultado** distintos reportes de vulnerabilidades realizados con herramientas como Nessus, OpenVas o Qualys en una sola aplicación desde la cual se

pueda llevar la gestión y seguimiento de las vulnerabilidades por parte el equipo de Security Operations.

2. **Priorizar** la gestión de vulnerabilidades siguientes los siguiendo los estándares de CVSS, realizando el cálculo de la criticidad y riesgo para cada vulnerabilidad acorde a los valores de la clasificación de la información de los activos gestionados.
3. Disponer de tablas de **seguimiento** en la que se puedan asignar las vulnerabilidades a los usuarios y que éstos puedan ir modificando su estado en las distintas fases.
4. Generación de **reportes automatizados** para su posterior entrega al cliente.
5. **Reportes personalizados** para la gestión que realiza el departamento para cada uno de los clientes.
6. Consulta de **métrica que mide el nivel de seguridad** de los sistemas.

1.3 Estructura del documento

La memoria consta de los siguientes secciones:

- Introducción: indica la motivación y objetivos de este proyecto, así como este apartado donde se explica la estructura del documento.
- Descripción del problema: Resume el por qué de este proyecto y los motivos que llevaron a que fuera necesario el desarrollo de una aplicación que diera solución a diversos problemas en la gestión de vulnerabilidades del departamento de Seguridad.
- Análisis de requisitos: Recoge todos los requisitos recopilados durante múltiples reuniones con el personal del departamento. Es fundamental para entender el producto final.
- Solución desarrollada: resume el proyecto realizado dividido en módulos, indicando brevemente la funcionalidad de cada una de sus partes y en qué módulo se van cumpliendo cada uno de los requisitos.
- Integración y pruebas: indicación de los tipos de pruebas realizadas así como los distintos entornos en los que se ha desarrollado, probado e implementado la aplicación.
- Conclusiones y trabajo futuro: se presentan los objetivos logrados con el desarrollo de la aplicación así como futuras mejoras de la misma en futuras iteraciones del producto.
- Anexo Herramientas y lenguajes usados: se mencionan las tecnologías y lenguajes principales empleados en este proyecto.
- Anexo Estructura de ficheros: describe la estructura a modo de ejemplo de algunos de los ficheros que se emplean en la aplicación como entrada de datos.

2 DESCRIPCIÓN DEL PROBLEMA

En esta sección se realiza un planteamiento del problema, así como la definición de conceptos y el proceso de gestión de vulnerabilidades, que es en lo que se basa la solución planteada en esta memoria.

2.1 Conceptos

A continuación se definen algunos conceptos relacionados con el ámbito de este documento:

- **Amenaza:** Acción que dispone de potencial para causar daño.
- **Activo:** Elemento conectado a la red que proporciona algún servicio a usuarios u otros elementos de la red.
- **Vulnerabilidad:** Este concepto está relacionado con la amenaza y activo, ya que es la materialización de una amenaza en un activo en concreto.
- **Criticidad:**Valor que se le da a una amenaza de 0 a 10 teniendo en cuenta los valores de los vectores CVSS
- **Riesgo:** Combinación de la probabilidad de que una vulnerabilidad sea explotada en un activo y el impacto que éste pueda tener.
- **KPI:** Mide el nivel de securización de un conjunto de activos. Se trata de una fórmula que otorga un peso a cada vulnerabilidad latente en un activo según su riesgo (altas, medias o bajas) se pondera y da un valor entre 0 y 100 siendo este último el valor más alto, indicando que un activo se encuentra en un nivel alto de securización. El valor objetivo para considerar que un activo tiene un buen nivel de seguridad está fijado en un porcentaje mayor que 75%.
- **Clasificación de la información:** se define mediante la triada CID (Confidencialidad /Integridad/Disponibilidad), controles que garantizan la seguridad de un sistema siempre y cuando se preserven los tres.
  - Confidencialidad: término que previene la que la información sea empleada por personas o sistemas no autorizados.
  - Integridad: término que asegura mantener que los datos accedidos no hayan sido modificados o alterados.
  - Disponibilidad: término que asegura que la información contenida en un sistema es posible acceder a ella por las personas que así lo requieran y se encuentren autorizadas.
- **Vector CVSS:** es un vector con distintas métricas que como resultado final proporciona un valor entre 0 y 10 que indica la criticidad y riesgo de una vulnerabilidad. Este valor permite a las personas encargadas de la gestión de vulnerabilidades de un sistema priorizar la respuesta y recursos acorde al valor de este vector.

2.2 Investigación y planteamiento del problema

En el departamento de Security Operations la gestión de vulnerabilidades se realizaba en un documento Excel por cada cliente. Estos documentos se encontraban en una carpeta compartida en la que cualquier persona del departamento podía acceder. La gestión de cada uno de estos documentos daba lugar a distintos problemas, ya que cada uno tenía un formato distinto y los datos se relacionaban con los extraídos de la CMDB de forma manual.

El uso de documentos Excel para manejo de volúmenes grandes de datos hacía que en muchas ocasiones éste se bloquease, que los tiempos de carga fueran elevados y el riesgo de errores al gestionar tantos datos aumentase. A todo esto se sumaba que solo una persona podía estar trabajando en el mismo documento a la vez.

Con el trabajo diario en el departamento se podían comprobar todos los problemas existentes, por lo que se tomó la decisión de plantear la realización de una aplicación que diera solución a los siguientes problemas encontrados:

- Como cada cliente puede emplear un *scan* de vulnerabilidades distinto, esto provocaba que la salida de datos también lo fuera y por tanto la conversión de los datos e integración con los de la CMDB o seguimiento de vulnerabilidades que ya estaban en curso, hacía que el proceso fuera prácticamente manual. En este proceso de normalización y actualización de seguimiento se tardaba una media de 8h por cada escaneo de vulnerabilidades de cada cliente.
- Una vez los datos estaban en un formato Excel estandarizado y legible, no podían trabajar varias personas a la vez en el mismo documento, éste cuando tenía grandes volúmenes de datos y gráficas se bloqueaba, con la posibilidad de pérdida de datos o corrupción del documento.
- A cada cliente se le entrega un reporte sobre la gestión de vulnerabilidades en sus activos con una determinada periodicidad. Para la realización de este reporte se empleaban las tablas de datos y gráficas de Excel, copiando y pegando a un documento Word, este proceso aunque se iban automatizando algunas gráficas, se tardaba una media de 6h por cada reporte.
- En los reportes de entrega al cliente se incluyen datos históricos, ya que representan de una forma visual la evolución y mejoras gracias al proceso de gestión de vulnerabilidades. Estos datos se iban guardando de forma manual en tablas de Excel el día 1 de cada mes, en ocasiones daba lugar a errores al calcularlos u olvidos a la hora de introducirlos mensualmente. En este proceso se tardaba una media de 15 minutos.

En todos los puntos mencionados, se puede observar que la mayor parte del tiempo se dedicaba a tareas que es posible automatizar, reduciendo los tiempos y eliminando el error humano al manejar volúmenes grandes de datos.

En un primer lugar se planteó la idea de realizar un análisis de los productos disponibles en el mercado, pero tras dicho análisis y evaluación se pudo comprobar que no existía nada que cubriese todos los objetivos mencionados en el apartado 1.2. Esto es debido a que se necesitaba una aplicación con unos criterios muy concretos, que pudiese integrarse con otras

herramientas de la empresa. Por ello se optó por la decisión del desarrollo interno de una aplicación, que podía adaptarse completamente a los requisitos y ser moldeable en el futuro.

Con la aplicación desarrollada para la gestión de vulnerabilidades, se tuvo que redefinir todo el procedimiento de gestión establecido en la compañía, que queda resumido en el siguiente punto de esta sección para el mejor entendimiento de la aplicación en las sucesivas secciones.

2.3 Proceso de gestión de vulnerabilidades

La gestión de vulnerabilidades es un proceso cíclico y continuo que cubre la detección, evaluación y corrección de vulnerabilidades en los sistemas y aplicaciones gestionados por T-Systems. En este proceso no se evalúa únicamente la vulnerabilidad, sino que se tiene en cuenta la clasificación de la información de los activos y se clasifican las vulnerabilidades acorde a su nivel de riesgo empleando las métricas definidas en el estándar CVSS.

Para llevar a cabo este proceso de gestión de vulnerabilidades, es necesario disponer de:

- Inventario de activos. Incluye todos los elementos de los que se quiera gestionar las vulnerabilidades (elementos de infraestructura de red, servidores, impresoras…) de los cuales se debe disponer de la clasificación de la información para el cálculo del riesgo.
- Herramienta *scan* de vulnerabilidades. Como Nessus, OpenVas o Qualys se obtienen los resultados de vulnerabilidades en los activos.

En la siguiente subsección se describen las fases de este proceso que han sido definidas en T-Systems.

2.3.1 Etapas

Cuando se va a comenzar un nuevo proceso de gestión de vulnerabilidades con un cliente o conjunto de activos, la figura de CuSM elabora una planificación inicial de escaneo de vulnerabilidades anual en la que se define lo siguiente:

- **Grupos de escaneo:** Dependiendo del valor de los activos, se realiza un análisis de sus aspectos y funcionalidades para dividir los activos en distintos grupos, de tal modo que ante posibles incidencias provocadas por el escáner se minimice el impacto en la medida de lo posible.
- **Periodicidad:** Se define cada cuánto tiempo se escanean los distintos grupos así como las fechas/horas con menor impacto para el negocio del cliente.
- **Grado de intrusión:** Los escaneos de vulnerabilidades pueden configurarse con mayor o menos grado de intrusión, según lo definido por el cliente se pacta el tipo de escaneo. A mayor grado de intrusión, mayor probabilidad de un posible incidente.
- **Herramienta a emplear:** Dependiendo de cada cliente y se pueden emplear distintas herramientas como Nessus, Qualys u OpenVas.

Una vez se tienen definidos los puntos anteriores, se presentan a OPM y SDM para su aprobación y dan comienzo de las etapas del proceso, que se muestra en la siguiente figura:

Figura 1. Ciclo de vida del proceso de gestión de vulnerabilidades

Descubrimiento

En esta primera etapa, con la planificación definida por el CuSM se abre un ticket de tipo cambio en la herramienta SM9, en la que se planean las acciones a realizar y se sigue todo el proceso ITIL de buenas prácticas. Una vez que el cambio es confirmado por todos los implicados y se aprueba en el CAB, se programa en la sonda y se obtiene el informe vulnerabilidades. Estos datos se introducen la herramienta de gestión de vulnerabilidades.

Normalización

Debido a que se emplean diferentes herramientas para obtener la información de las vulnerabilidades en los sistemas, es necesario normalizar los datos para su posterior análisis y tratamiento. Para ello se sube el resultado del escaneo a la herramienta de gestión de vulnerabilidades, el cual normaliza los datos y los introduce en la base de datos para su posterior análisis.

Priorización y análisis

Una vez que la información se encuentra normalizada, la herramienta de gestión de vulnerabilidades proporciona un valor de criticidad a cada vulnerabilidad, que es el proporcionado por el resultado del escaneo, y un valor de riesgo, que lo obtiene empleando los cálculos definidos en el estándar CVSS en que tiene en cuenta el valor de la amenaza y el valor del activo al que afecta.

Siguiendo el valor del riesgo de cada vulnerabilidad comienza el análisis[1] de las vulnerabilidades por parte del equipo de seguridad, que determinan las acciones a realizar con la finalidad de mitigar o solventar el riesgo.

[1] El proceso de análisis de cada vulnerabilidad requiere de conocimientos técnicos por parte del equipo de seguridad y dependiendo de cada caso se determinan las acciones más recomendables a realizar.

En esta fase de análisis, si se determina que es posible remediar o mitigar las vulnerabilidades, se abre un ticket de tipo problema en la herramienta SM9 en el que se contempla todo el análisis y las implicaciones del resto de grupos técnicos.

En algunas ocasiones las vulnerabilidades no pueden ser resueltas, como por ejemplo debido a incompatibilidades con aplicaciones del propio cliente o porque la relación coste/beneficio no compensa al cliente. En estos casos se redacta una aceptación del riesgo que debe ser firmada por el cliente.

Las decisiones tomas en la fase de análisis quedan reflejadas en la herramienta de gestión de vulnerabilidades por parte de los analistas de seguridad.

Planificación

Para realizar una intervención en un activo, se debe coordinar la intervención teniendo en cuenta a los equipos técnicos, a los propietarios de los activos así como al cliente, ya que éste último es el que nos proporciona una ventana de actuación. Una vez se validan las acciones a realizar y se acuerda una ventana de actuación se procede a la siguiente fase.

Remediación /Mitigación

En la ventana de actuación pactada, se realizan las acciones para solventar o mitigar las vulnerabilidades encontradas. Esta información queda reflejada en la herramienta de gestión de vulnerabilidades.

Verificación

El proceso de gestión de vulnerabilidades es cíclico, por lo que una vez realizadas las acciones correctivas, se procede a lanzar de nuevo un escaneo de vulnerabilidades para comprobar que se ha implementado correctamente. De forma periódica se va repitiendo el ciclo.

3 ANÁLISIS DE REQUISITOS

En esta sección se presentan los requisitos recogidos para la implementación de la aplicación. En esta fase del proyecto se mantuvieron múltiples reuniones con las distintas personas del departamento de Security Operations. Esta fase fue primordial para conocer todas las carencias del proceso así como conocer las necesidades de los técnicos que llevaban en el proceso más tiempo, al igual que del mánager que requería de requisitos adicionales.

A continuación se describe la funcionalidad y características de la aplicación en forma de análisis de requisitos funcionales y no funcionales que se extrajeron de las múltiples reuniones con los miembros del departamento.

3.1 Requisitos funcionales

Los requisitos funcionales definen la funcionalidad que debe desempeñar el software. Éstos describen los comportamientos esperados del sistema para los distintos casos de uso.

3.1.1 Gestión de usuarios

- **RF001:** para acceder a la aplicación se debe realizar a través de un formulario de inicio de sesión.
- **RF002:** La contraseña de cada usuario deberá poder modificarse desde el perfil de cada uno.
- **RF003:** No se permitirá el borrado de usuarios, pero sí deshabilitar el acceso a la aplicación.
- **RF004:** Solo los usuarios administradores podrán cambiar la contraseña del resto de usuarios.

3.1.2 Subida de scans

Los *scans* vulnerabilidades pueden proceder de distintas fuentes, por ello la aplicación debe permitir seleccionar qué entrada de datos se le está dando.

- **RF005:** Se debe permitir seleccionar con un desplegable la herramienta de la cual proviene el resultado del escaneo para poder normalizar los datos.
- **RF006:** En la base de datos se debe almacenar toda la información que dispone el fichero, los campos imprescindibles son cliente, ip, puerto, datos de la vulnerabilidad y fecha del scan.

3.1.3 Gestión de servidores

En T-Systems se dispone de una aplicación denominada CMDB en la que se dispone de toda la información relativa a los activos que administra o gestiona la empresa. Es una base de datos extensa y se emplea para conocer el estado de todos los activos. La CMDB es muy

lenta, por lo que se requiere extraer la información necesaria para la gestión de vulnerabilidades y que se encuentre disponible en la aplicación a desarrollar.

- **RF007:** Se deber cargar en la aplicación la información relativa a los servidores a través de un fichero extraído de la CMDB.
- **RF008:** La aplicación debe disponer de un apartado para subir este fichero por cualquier usuario, partiendo de un fichero normalizado. La estructura de este fichero se detalla en el anexo B.1.
- **RF009:** La información disponible en la CMDB de los servidores no podrá ser modificable en la aplicación.
- **RF010:** Se debe disponer de la información de servicio, entorno, estado, ip's y función de red de cada uno de los servidores.
- **RF011:** Como cada servidor debe disponer de los valores de Confidencialidad, Integridad y Disponibilidad. Por defecto vendrán como en estado no definido, y se podrán dar valores para el cálculo del riesgo de las vulnerabilidades según el estándar CVSS.
- **RF012:** Se debe almacenar la fecha en la que se ha subido el último scan de vulnerabilidades de cada uno de los servidores.

3.1.4 Gestión de amenazas

- **RF013:** La aplicación debe disponer de una tabla en la que se pueda consultar toda la información de las amenazas de las distintas fuentes de datos. Esta información debe estar disponible para todos los usuarios autenticados de la aplicación.
- **RF014:** Con cada reporte de escaneo de vulnerabilidades nuevo que suba a la aplicación, se debe actualizar la información de las amenazas.
- **RF015:** La aplicación debe gestionar amenazas con múltiples CVE's.
- **RF016:** Se debe almacenar el vector de CVSS de cada vulnerabilidad.

3.1.5 Gestión de vulnerabilidades

Uno de los principales objetivos de la aplicación es poder hacer una gestión de vulnerabilidades eficiente, pudiendo priorizar según cliente y riesgo de las vulnerabilidades. Además, es imprescindible poder hacer seguimiento de cada una de las vulnerabilidades, conocer quién la está gestionando y el estado de la misma.

A continuación se recogen los requisitos para esta sección:

- **RF017:** Se considerarán vulnerabilidades iguales cuando coincida: cliente/nombre servidor/puerto/id de amenaza.
- **RF018:** Se le podrá asignar a las vulnerabilidades los siguientes estados:
    - Review: aún no se ha hecho ningún tratamiento a la vulnerabilidad encontrada por parte de Security Operations.

- Pending SDM/OPM: pendiente de aprobación por parte de los responsables del servicio.
- Pending client: pendiente de aprobación de cliente.
- PendingGroup: pendiente de acciones por parte de los equipos técnicos que implementan la solución.
- Pending PM: Pendiente de abrir ticket en la herramienta de tickets por parte de Security Operations.
- Solved: vulnerabilidad resuelta.
- Accepted: vulnerabilidad con el riesgo aceptado, debido a que es un falso positivo o una alegación del cliente.
- Reopen: cuando una vulnerabilidad resuelta en el pasado, vuelve a aparecer.

- **RF019:** Las vulnerabilidades deben poder priorizarse en cuanto a criticidad y riesgo.
- **RF020:** Las vulnerabilidades deben poder asignarse a usuarios de la aplicación.
- **RF021:** Las vulnerabilidades deben disponer de la fecha inicial de detección y la última vez que han sido detectadas.
- **RF022:** Se deben poder modificar varios registros de vulnerabilidades a la vez.
- **RF023:** Los usuarios no pueden borrar vulnerabilidades de la base de datos.
- **RF024:** Las vulnerabilidades de tipo INFO no se incluirán en la base de datos de la aplicación, ya que no suponen un riesgo para los servidores.

3.1.6 Gestión de aceptaciones

En el proceso de gestión de vulnerabilidades, éstas no siempre se pueden solventar. Se diferencian 2 motivos:

- **Alegación:** Una vulnerabilidad se considera en estado de alegada cuando ésta no va a ser resuelta. Los motivos pueden ser diversos: la relación coste-riesgo no quiere ser asumida por el cliente, sistemas operativos obsoletos que no se pueden actualizar debido a incompatibilidades con software del cliente necesario para su actividad, se mitiga el riesgo con algunas acciones pero éste sigue sin ser cero…etc. En estos casos, siempre se redacta un documento de aceptación de riesgo que se entrega a cliente y éste lo firma asumiendo el riesgo de no aplicar una solución.
- **Falso positivo:** en ocasiones se ha aplicado la configuración correcta, o se ha aplicado un *workaround*[2], para solventar una vulnerabilidad. En estos casos, la vulnerabilidad sigue apareciendo como activa al pasar un scan de vulnerabilidades, pero se trata de un falso positivo.

[2] Solución temporal que se emplea ante la imposibilidad de aplicar la definitiva.

Cuando una vulnerabilidad se encuentra en estado aceptado, no cambiarán su estado cuando vuelvan a aparecer en nuevos scans, y quedan fuera de la gestión de vulnerabilidades hasta que el cliente indique lo contrario.

- **RF025:** Se debe mantener un registro con todas las aceptaciones de cada cliente, en el cual se incluya si ésta se encuentra firmada o no.

3.1.7 Cálculo de KPI

El cálculo del KPI se emplea para medir el nivel de riesgo al que se encuentran expuestos los sistemas. Los requisitos relativos a este KPI son los siguientes:

- **RF026:** El cálculo del KPI total debe estar disponible en la aplicación con los datos de ese mismo momento
- **RF027:** Se podrá consultar el KPI por cliente.
- **RF028:** Se guardará un histórico mensual del KPI del día 1 de cada mes.

3.1.8 Generación de reportes

La parte más importante, a parte de la gestión de vulnerabilidades, es la generación de reportes que se le entregan a cliente. Con los reportes, se pone en valor todo el trabajo realizado y el cliente puede medir en qué grado se están cumpliendo sus expectativas con el servicio contratado. Hasta ahora todos los reportes se hacían a mano, haciendo de esta tarea un trabajo tedioso, con posibles errores y aproximadamente entre 6h y 2 días de trabajo dependiendo del cliente. Por ello los requisitos de éste módulo son de una importancia alta.

- **RF029:** los reportes deben poder sacarse por cliente y con filtrado de fechas de inicio y fin.
- **RF030:** los reportes deben tener los siguientes apartados:
  - Planificación de scans (scans realizados en el periodo del reporte y próximos scans de vulnerabilidades)
  - Seguimiento de tickets (tickets abiertos y cerrados en el periodo del reporte, y seguimiento de antiguos)
  - TOP 10 de vulnerabilidades comunes y críticas
  - Cantidad de vulnerabilidades latentes por servidor
  - Posibilidad de sacar reportes teniendo en cuenta criticidad y riesgo o solo criticidad.
  - Documentos de alegación por cada cliente así como falsos positivos.
  - Alcance del reporte y grupos de escaneo de los servidores.
- **RF031:** Los reportes deben ser en formato Word.

3.1.9 Gestión de tickets

En T-Systems, cualquier cambio o modificación que se realice en cualquiera de los servidores que en encuentran en la CMDB en estado Operativo, debe documentarse en la

herramienta de *ticketing* SM9. Estos tickets siguen un riguroso proceso de calidad siguiendo la metodología ITIL.

- **RF032:** Todos los cambios relacionados con scan de vulnerabilidades deben poder incluirse en tablas en la aplicación y especificarlo los servidores afectados.
- **RF033:** Todos los problemas relacionados con la solución de vulnerabilidades deben poder incluirse en la aplicación, de tal modo que se pueda hacer seguimiento de los mismos, y asignarlos a vulnerabilidades.

3.2 Requisitos no funcionales

En esta sección se recogen los requisitos no funcionales de la aplicación, es decir, aquellos que no hacen referencia a la funcionalidad o comportamiento.

- **RNF001:** El cálculo del KPI global debe ser inferior a 30 segundos.
- **RNF002:** La aplicación debe ser práctica para los miembros del departamento.
- **RNF003:** La aplicación debe emplear las fuentes y los colores establecidos por T-Systems.
- **RNF004:** la generación de reportes para clientes con hasta 300 activos debe ser inferior a 1 minuto.
- **RNF005:** la aplicación debe permitir el uso de múltiples usuarios simultáneamente.
- **RNF006:** la aplicación debe poder alojarse en servidores Linux (RedHat y Oracle Linux).
- **RNF007:** La base de datos debe ser MySQL.
- **RNF008:** La aplicación debe desarrollarse en PHP principalmente.
- **RNF009:** Los reportes de gestión de vulnerabilidades deben generarse en menos de 3 minutos.
- **RF010:** Las búsquedas en las tablas de datos deben ser inferiores a 2 segundos.

4 SOLUCIÓN DESARROLLADA

La interfaz de usuario de la aplicación desarrollada se diseñó con la finalidad de ser intuitiva. Para ello se siguieron las líneas de diseño de otras herramientas del departamento, de este modo la curva de aprendizaje inicial se reduciría. Los prototipos iniciales fueron evolucionando y mejorando al igual que sufrieron alguna modificación adicional en las fases de validación de los usuarios.

Para acceder a la aplicación es necesario disponer de un nombre de usuario y contraseña que se le proporciona a cada usuario. Una vez que se ha iniciado sesión, se accede a la pantalla principal donde se muestran datos generales de la gestión de vulnerabilidades: los evolutivos de KPI, cantidad total de vulnerabilidades, riesgo acumulado, los top 10 de vulnerabilidades comunes así como las críticas (RF026, RF027, RF028).

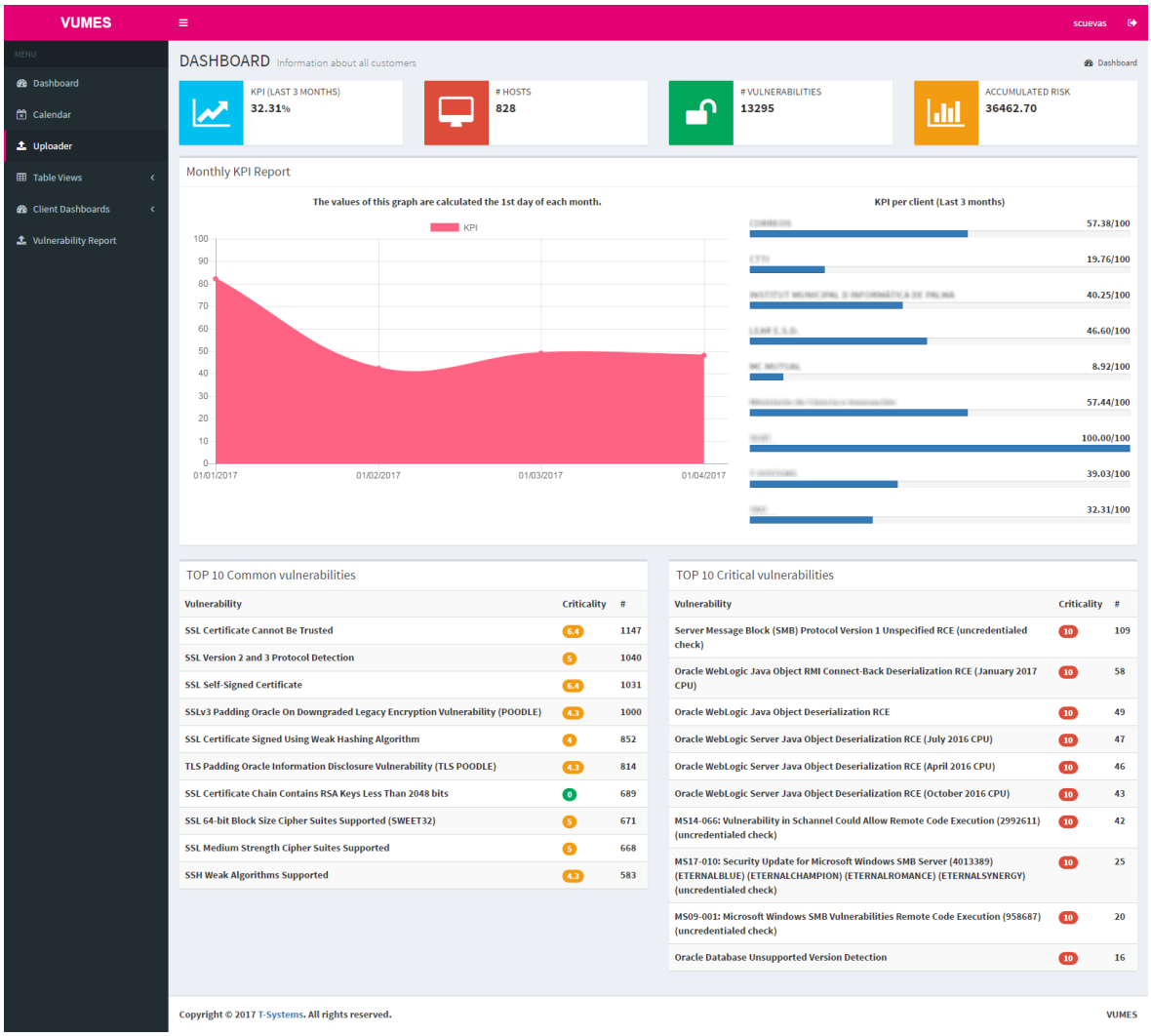

Figura 2. Pantalla al inicio de sesión

En la parte superior derecha se muestra el nombre de usuario, donde si se hace clic se podrá modificar la contraseña del mismo (según el perfil de usuario se cumplen en los requisitos RF001, RF002, RF003, RF004). También se encuentra el botón de cierre de sesión para salir de la aplicación.

El usuario puede navegar por el menú lateral de la aplicación, por las distintas secciones que se muestran en la Figura 3:

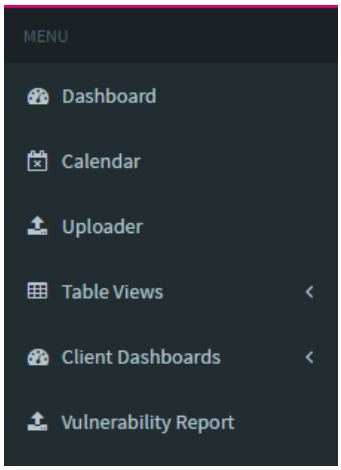

Figura 3. Menú lateral de navegación

A continuación vamos dividir la solución de la aplicación en sus distintos módulos, en los cuales se detalla su funcionalidad así como las decisiones de diseño tomadas para cumplir con los requisitos descritos en el punto 3 de este documento.

- **Módulo de carga de datos**: se gestiona la carga de datos masiva proveniente de la CMDB o de *scans* de vulnerabilidades. Mediante un formulario el usuario puede subir los datos a la aplicación.
- **Módulo de tablas**: se trata de distintas vistas de tablas de la base de datos (vulnerabilidades, amenazas, servicios, servidores, tickets…) con los que el usuario interacciona para llevar a cabo la gestión y seguimiento de las vulnerabilidades.
- **Módulo de calendario**: muestra de forma visual la planificación de *scans* de vulnerabilidades.
- **Módulo de reportes**: se encarga de mediante un formulario que rellena el usuario, de la generación de reportes por cliente en un formato Word.
- **Módulo de *dashboards***[3]: en estas pantallas se encuentra información general de la gestión de vulnerabilidades, gráficos de KPI, cantidad de vulnerabilidades, hosts...etc. Con ello se da una idea general del trabajo que queda por realizar, si van bien los objetivos marcados y el estado general del servicio.

4.1 Módulo de carga de datos

Los datos que recibe la aplicación se pueden dividir en dos tipos: los obtenidos de la CMDB y los obtenidos de un reporte de escaneo de vulnerabilidades. Estos datos son cargados a la aplicación a través de un formulario que debe rellenar el usuario (RF005). La este módulo se encarga de validar y normalizar todos los datos para introducirlos en las distintas tablas de la

[3] Se refiere a pantallas de la aplicación en las que se dispone de forma gráfica distintas gráficas y tablas que proporcionan información general del estado de la seguridad de un cliente o conjunto de ellos.

base de datos (RF006). La estructura de algunos de estos ficheros se encuentra detallada en el anexo B.

4.1.1 Datos de CMDB

La CMDB es una aplicación en la que se mantiene actualizada toda la información relativa a los activos gestionados por T-Systems. Esta aplicación es muy lenta y uno de los requisitos de este proyecto es que los datos se cargasen directamente desde un fichero que se extrae semanalmente de la CMDB con los datos necesarios y no conectando ambas aplicaciones (RF007).

El detalle de la estructura del fichero de la CMDB que se emplea para introducir los datos en la aplicación se detalla en el anexo B.1. La información de este fichero se compara con la contenida en la base de datos, añadiendo, eliminando y modificando posibles cambios.

Una vez subido el fichero, se indica al usuario si se ha subido correctamente o si ha encontrado algún problema en los datos. (En este módulo se cumplen los requisitos RF008, RF009, RF010, RF011, RF012).

4.1.2 Datos de scans de vulnerabilidades

Los *scans* de vulnerabilidades se pueden realizar con distintas herramientas dependiendo del cliente, por ello en el diseño y desarrollo de este módulo se realizó una función para cada tipo de fichero, ya que cada uno dispone de una configuración diferente. La estructura de algunos de estos ficheros se detalla en el anexo B.2.

En este módulo se extrae toda información necesaria de los ficheros y se normaliza para ser introducida en la base de datos. Este paso de normalización es de los más importantes ya que permite que los usuarios desde una única vista en la aplicación web puedan hacer la gestión de vulnerabilidades y generación de reportes igual para todos los clientes independientemente de la herramienta empleada para los escaneos.

En este paso de normalización de datos de los scans, para las vulnerabilidades se realiza el cálculo del riesgo empleando el vector CVSS proporcionado en el reporte del scan y los datos contenidos en la base de datos sobre la confidenciad, integridad y disponibilidad del activo al que afecta la vulnerabilidad. También se hace coincidir los datos del scan con los contenidos de la CMDB de tal modo que de cada IP escaneada se puede obtener toda la información que ofrece la CMDB. Si la IP escanada no se encuentra en la base de datos, ésta no se puede introducir en la base de datos y se muestra un mensaje de error al usuario para que revise el error (RF017, RF019, RF021, RF024).

Antes de introducir cada vulnerabilidad del scan en la base de datos se hace la comprobación de si ésta ya existe en la base de datos, ya que es posible que haya aparecido en anteriores scans y aún no haya sido solucionada. Se pueden dar los siguientes casos:

- Si la vulnerabilidad no ha aparecido antes, se introduce en la base de datos con la fecha de detección la del scan actual, así como la última fecha de modificación. El estado que se le da a esta vulnerabilidad es *Review*.
- Si la vulnerabilidad ya se encuentra en la base de datos, se comprueba su estado dado lugar a las siguientes opciones:

- Si el estado es *Solved,* quiere decir que la vulnerabilidad que se daba por solucionada ha vuelto a aparecer, por lo que se cambia su estado a Reopen y se actualiza la última fecha de modificación.
- Si el estado es *Accepted,* es una vulnerabilidad que se aceptó el riesgo y por tanto va a seguir apareciendo, por lo que no se modifica su estado, pero sí su última fecha de modificación, para que el usuario pueda saber cuando fue la última vez que apareció.
- Para cualquier otro estado, solo se actualiza la fecha de modificación.

En este paso de normalización, también se actualiza la base de datos de amenazas, con toda la información que proporciona el scan, como id, nombre, descripción, CVE's…etc. (RF014, RF015, RF016)

Una vez la información se ha introducido en la herramienta y ésta la ha normalizado, el usuario puede acceder a las distintas tablas para gestionar las vulnerabilidades. Cada una de las tablas se encuentra paginada, dispone de un buscador global y uno por cada campo de la tabla. El usuario puede actualizar la información de cada vulnerabilidad y sus estados, así como asignarse vulnerabilidades para coordinar el trabajo con el resto del equipo.

4.2 Módulo de tablas

Una vez toda la información de CMBD y de vulnerabilidades se encuentra en la base de datos los usuarios pueden trabajar con esta información a través de la interfaz web con distintas tablas definidas.

Las tablas definidas para mostrar la información a los usuarios son las siguientes:

- **Scans:** Contiene todos los tickets abiertos en la herramienta SM9 relativos a escaneos de vulnerabilidades. En esta tabla se van introduciendo los escaneos de vulnerabilidades planificados de cada cliente y su seguimiento. Todos estos escaneos se muestran también en un calendario para poder identificar de forma gráfica los posibles solapamientos (RF032).
- **Problems:** Contiene todos los tickets de tipo "PM" abiertos en la herramienta SM9 que se emplean para la aplicación de soluciones a las vulnerabilidades por parte de los equipos técnicos. Los usuarios van introduciendo todos los tickets abiertos a los equipos técnicos para la resolución de las vulnerabilidades, así como su seguimiento (RF033).
- **Acceptances:** Todos los documentos que se redactan para la alegación o falsos positivos de vulnerabilidades de cada cliente y su estado (RF025).
- **Services**: Se muestra información de los servicios que pertenecen a un cliente y su estado. Los datos se obtienen de la CMDB. Esta tabla no es modificable por el usuario, pero proporciona información para la el proceso de gestión de vulnerabilidades que realizan los usuarios.
- **Environments:** Se muestra información de los entornos de un cliente y su estado. Al igual que la tabla anterior, no es modificable por el usuario.

- **Hosts:** Tabla en la que los datos son obtenidos de la CMDB. Los campos que se obtienen de ésta no son modificables por el usuario, pero sí puede añadir los datos de Confidencialidad, Integridad y Disponibilidad de acuerdo a los valores definidos en el estándar CVSS de cada servidor para posteriormente el cálculo del riesgo de cada vulnerabilidad. El usuario también puede añadir un grupo a cada servidor, que correspondería con el grupo de servidores que se escanea conjuntamente cuando hay que realizar un scan de vulnerabilidades planificado.
- **Vulnerabilities**: Contiene la información de todas las vulnerabilidades detectadas, su estado, servidor, servicio, entorno, descripción…etc. que es útil para el usuario a la hora de realizar la gestión y seguimiento. En esta tabla el usuario puede modificar los comentarios y el estado de cada vulnerabilidad (RF018, RF020, RF022, RF023).
- **Pending vulnerabilities:** Es igual a la tabla anterior, pero excluyendo las vulnerabilidades que ya se encuentran solucionadas o alegadas.
- **Threats:** Muestra la información de las amenazas detectadas en todos los escaneos. Se identifican por un ID único dependiendo de la tecnología y se muestra el nombre, descripción y posibles soluciones o *workarounds* para dicha amenaza (RF013).

Todas las tablas que se muestran en la aplicación disponen de las siguientes características:

- Paginación de resultados, haciendo una consulta por cada página que se pasa, de tal modo que las tablas cargas de forma instantánea.
- El orden de las columnas es intercambiable para que el usuario las ordene acorde a su conveniencia a la hora de trabajar con ellas.
- Búsqueda global así como búsquedas por columnas.
- No todos los campos son modificables por el usuario, pero los que sí es posible, el usuario puede hacer selecciones múltiples y modificaciones.

En la siguiente figura se muestra un ejemplo de tabla en el que se están editando los datosde la misma:

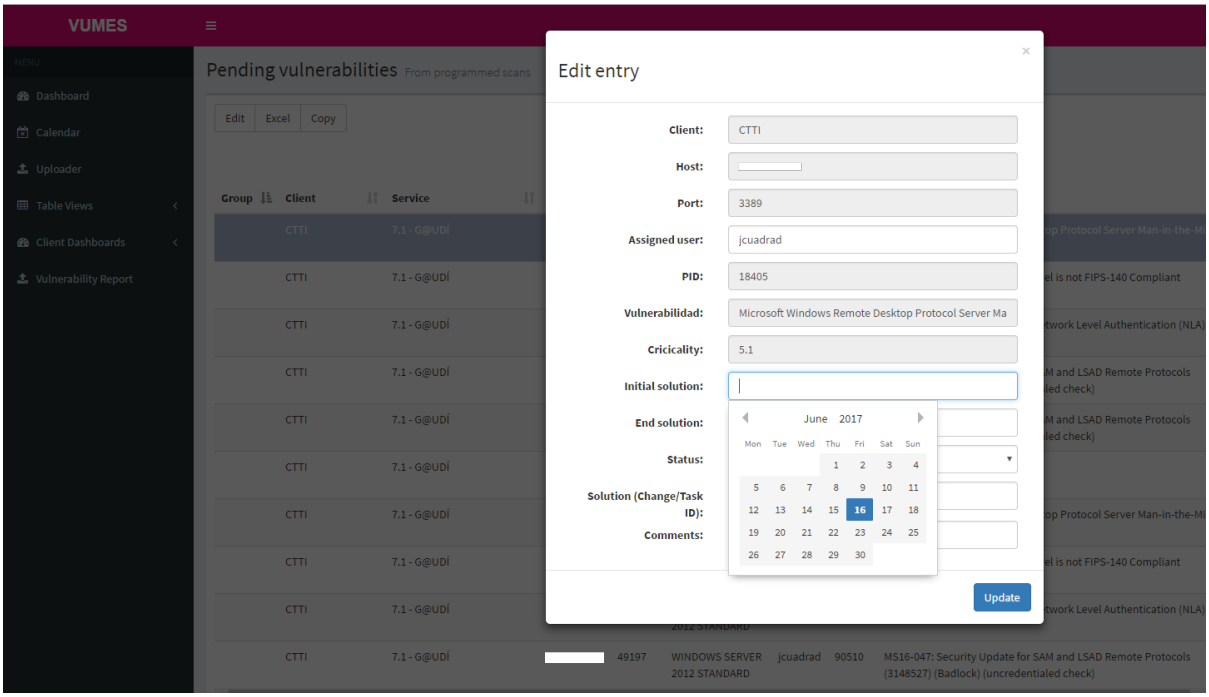

Figura 4. Edición de tabla

4.3 Módulo de calendario

La aplicación dispone de un calendario visual en el que se muestran todos los scans de vulnerabilidades programados para cada cliente, esto permite a los usuarios ver de un solo vistazo que hay planificado en una semana y detectar posibles anomalías. El usuario debe rellenar un formulario en el que se indica la fecha inicio, fecha fin, el cliente y el grupo de servidores afectados.

En la siguiente figura se muestra un ejemplo del mes de Mayo.

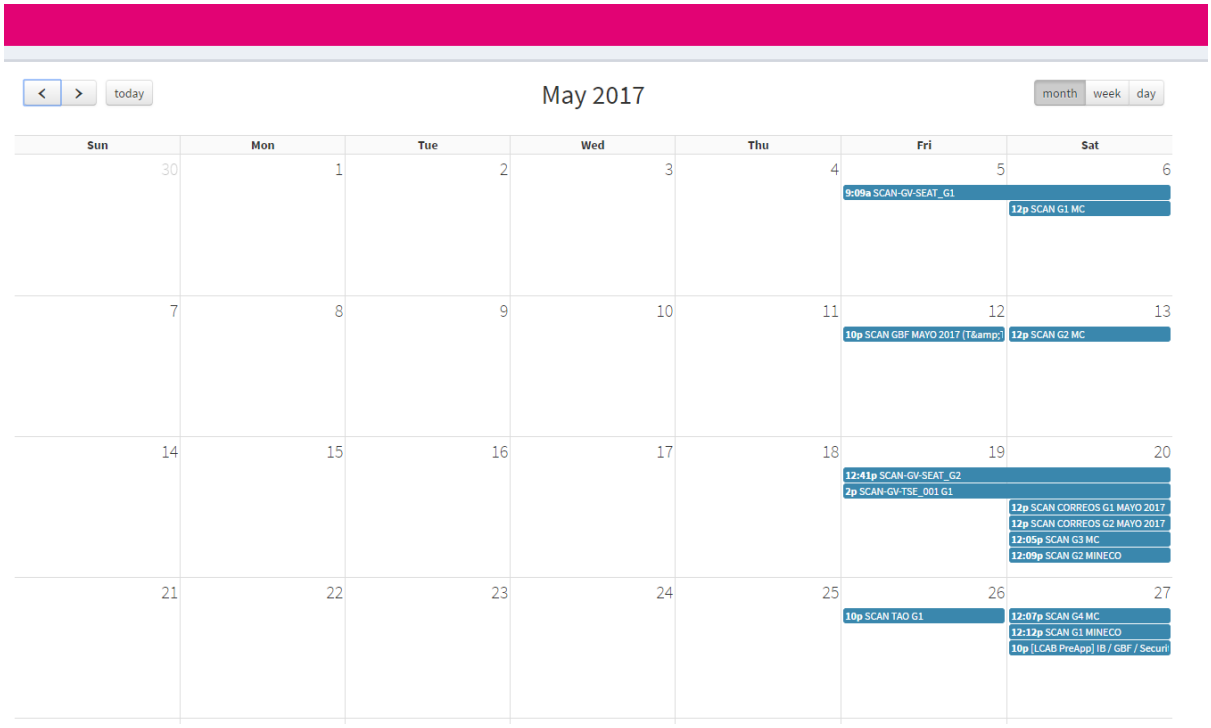

Figura 5. Módulo de calendario

4.4 Módulo de generación de reportes

Este módulo permite a los usuarios de la aplicación la generación de reportes en formato Word escogiendo cliente, fecha inicio y fecha fin y tipo de reporte de forma automática (RF029, RF031). En la siguiente imagen se puede ver el formulario a rellenar por el usuario.

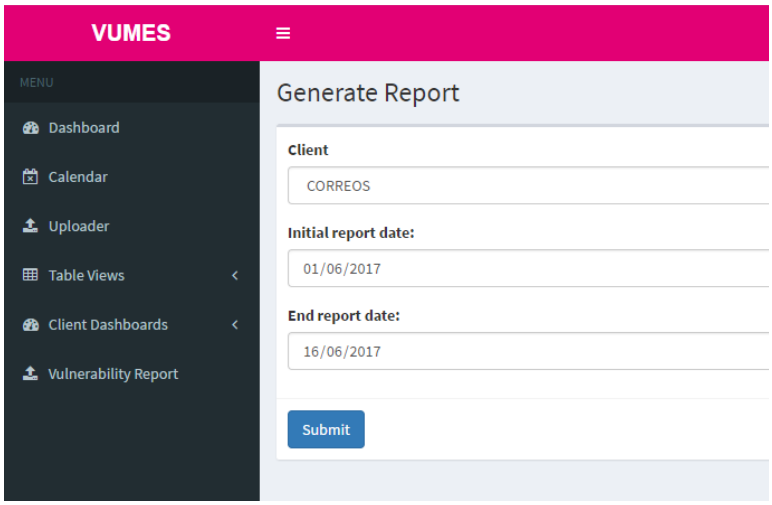

Figura 6. Generación de reporte en formato Word

Con este módulo se buscaba la estandarización a la hora de realizar reportes a los diferentes clientes por ello se establecieron únicamente 2 tipos de reportes, uno orientado a la criticidad y otro orientado al riesgo (RF030). No ha sido posible realizar un único reporte ya que para realizar un reporte de vulnerabilidades orientado al riesgo se debe conocer la clasificación de la información de los activos gestionados y en algunas ocasiones esta información no es proporcionada por el cliente. A pesar de que ésta daría un valor añadido a la gestión, obtener

la clasificación de la información en los activos de un cliente es necesario realizar un análisis de riesgos y conocer a fondo los servicios dados.

Independientemente del tipo de reporte que se genere con la aplicación, son bastante similares en cuanto a estructura, en los casos que se habla del riesgo, para los reportes que se generan sin clasificación de la información estos puntos no están incluidos. A grandes rasgos se componen de los siguientes elementos:

- Portada: En la que se indica el título, cliente al que aplica, y la fecha.
- Índice de contenidos: Contiene la estructura del documento.
- Introducción: Se indica el objetivo del reporte así como el alcance del mismo.
- Metodología: Describe todos los términos que se emplean el reporte así como la metodología seguida en el proceso.
- Gestión de vulnerabilidades: es el punto más amplio del documento en el que se detalla con los distintos sub-apartados la gestión que hay en proceso así como datos y estadísticas de la gestión de vulnerabilidades.
    - Planificación de escaneos: incluye una tabla con los escaneos realizados en el periodo del reporte y otra con los próximos escaneos planificados.
    - Planificación de problemas: varias tablas en las que se muestran los problemas abiertos a los equipos técnicos y su estado y los problemas cerrados durante el periodo del reporte.
    - Cantidad de vulnerabilidades latentes: Incluye una gráfica con las vulnerabilidades pendientes por resolver en términos de criticidad y riesgo. También se incluye una tabla con todos los servidores del alcance y la cantidad de vulnerabilidades que tiene cada uno según la criticidad/riesgo de tipo bajas (0-3.9) , medias (4- 6.9), altas (7-8.9) y críticas (9-10).
    - TOP 10 vulnerabilidades comunes y TOP 10 según criticidad/riesgo: incluye 2 gráficas de barras con el TOP 10 de cada caso así como 2 tablas con las vulnerabilidades de dichas gráficas.
    - Valor acumulado de la criticidad/riesgo: se muestran 2 gráficas con el valor acumulado por servidor y por servicio así como los mismo datos en formato de tabla de datos.
    - Vulnerabilidades latentes y servidores afectados: tabla con todas las vulnerabilidades pendientes de resolver y los servidores a los que afecta.
    - Vulnerabilidades nuevas: tabla de vulnerabilidades que hasta el momento del reporte no habían aparecido en anteriores ocasiones.
    - Aceptaciones: se incluye una tabla con todos los documentos de aceptación y su estado además de una tabla que contiene todas las vulnerabilidades de las cuales se encuentra aceptado el riesgo.

4.5 Módulo de dashboards

En este módulo se genera un panel con gráficas personalizadas para cada cliente que tiene servicio de gestión de vulnerabilidades. De este modo, el usuario encargado de la gestión puede visualizar en distintas gráficas el estado general de la seguridad, identificar los servidores más críticos, las vulnerabilidades más comunes para dicho cliente o medir la criticidad o riesgo agrupando por servicios.

Existe un panel genérico que se emplea cada vez que se añade un cliente nuevo, pero éstos se personalizan según las necesidades de cada cliente. En las siguientes imágenes se muestran algunas de las gráficas que aparecen en los paneles de cada cliente. Debido a que se trata de datos reales, algunos de los datos han sido ofuscados.

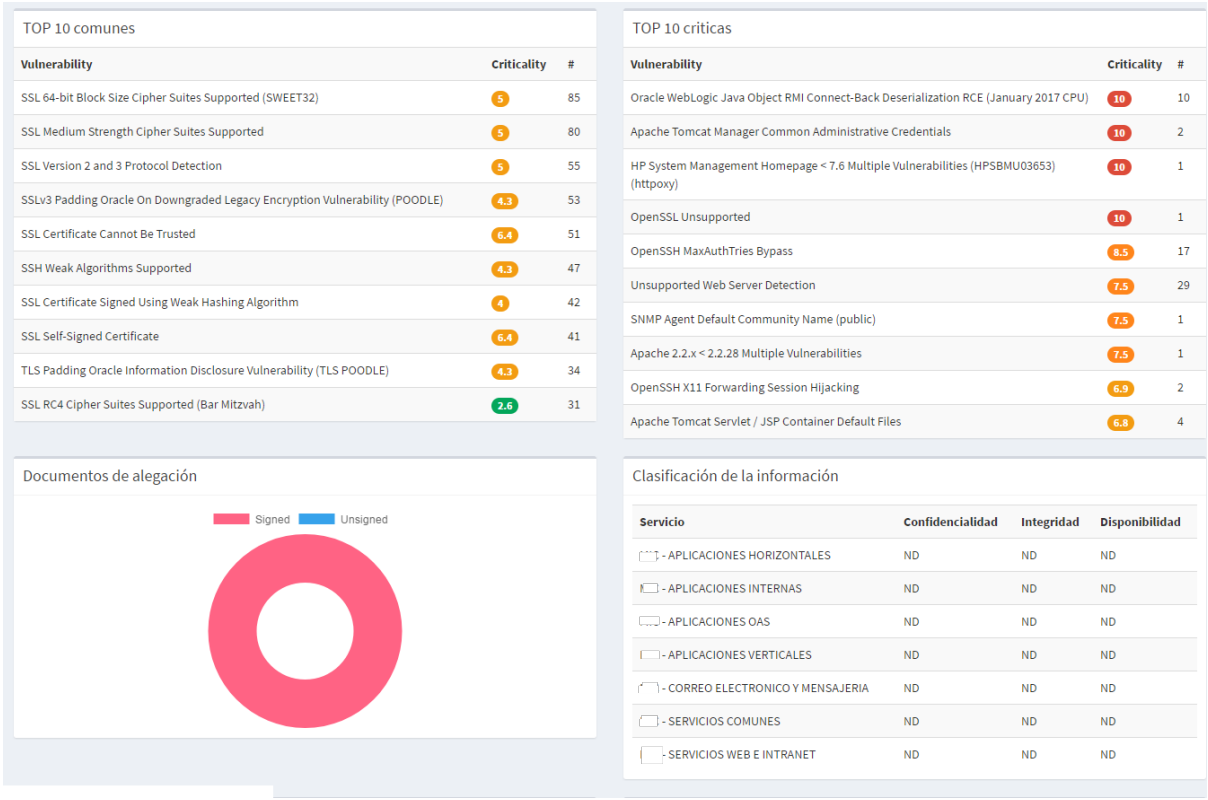

Figura 7. Ejemplo de dashboard 1

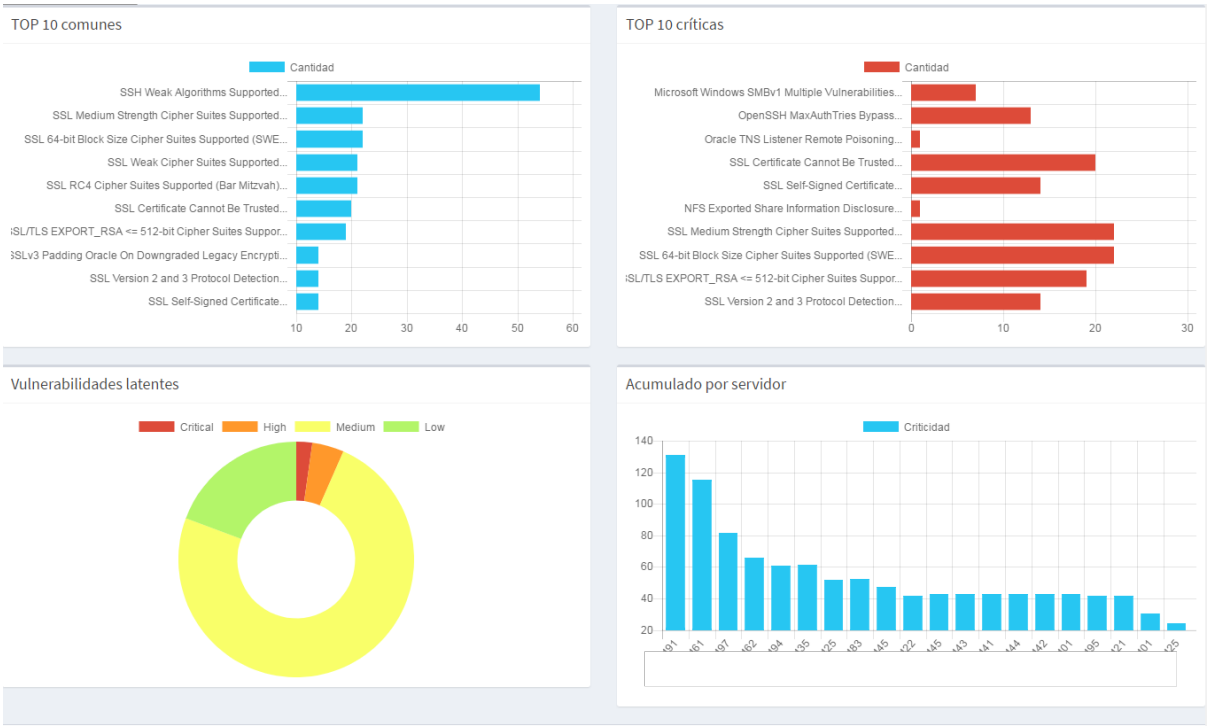

Figura 8. Ejemplo de dashboard 2

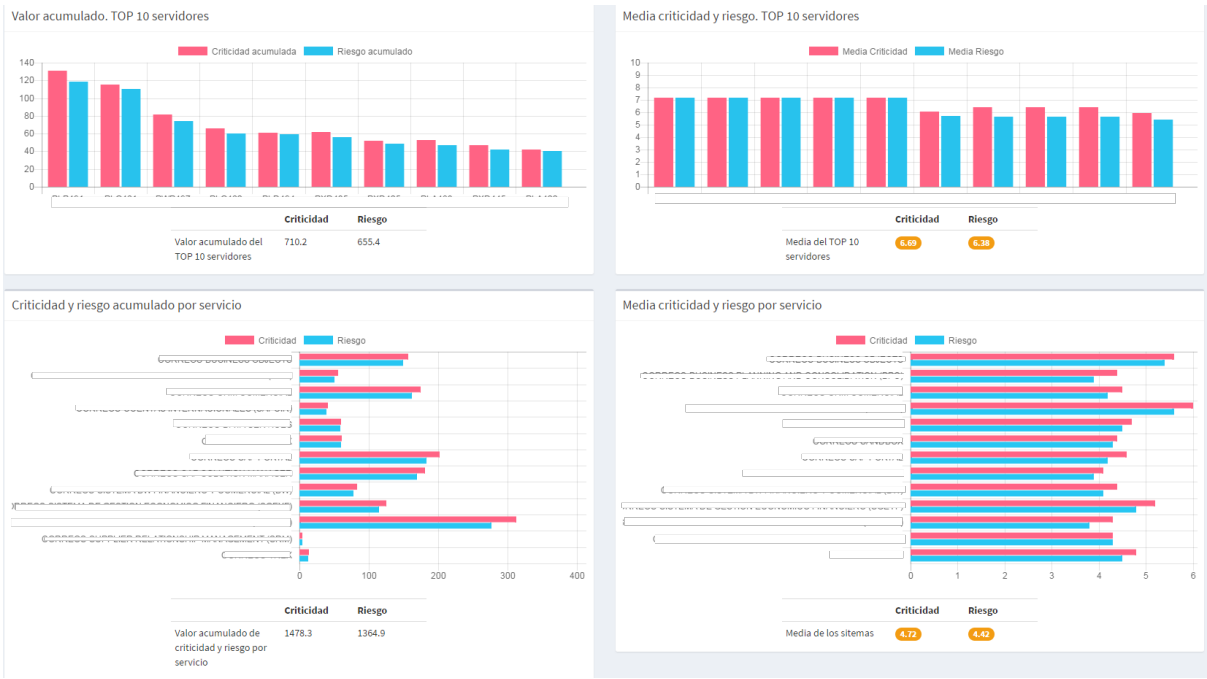

Figura 9. Ejemplo de dashboard 3

5 INTEGRACIÓN, PRUEBAS

Para comprobar el funcionamiento de esta aplicación se han realizado multitud de pruebas unitarias de cada uno de los módulos, pruebas de integración de los mismos y pruebas con usuarios reales de la aplicación.

En todas las pruebas realizadas cabe destacar que siempre se siguió el siguiente esquema de entornos, cada vez que se requería de nueva funcionalidad y desarrollos se seguía de izquierda a derecha:

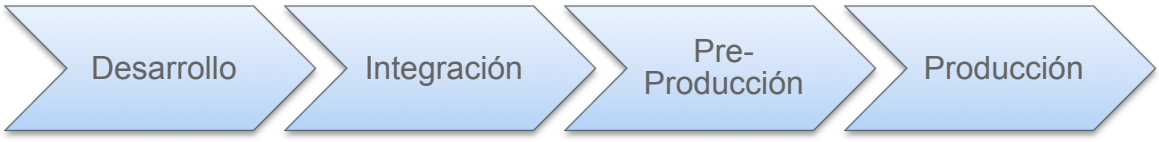

Figura 10. Fases del proceso de pruebas

- **El entorno de desarrollo** se emplea cuando se quiere realizar una nueva funcionalidad o módulo para la aplicación, los cambios en este entorno son constantes y el código evoluciona de forma rápida.
- **El entorno de integración** se emplea como su nombre indica, para la integración de los distintos módulos de la aplicación. En este entorno no se trabaja con datos reales.
- **En el entorno de pre-producción** no se desarrolla nada de código, en cuanto a código es exactamente igual que el entorno productivo. En este entorno se emplean datos reales para poder testear el funcionamiento de toda la aplicación y verificar que el comportamiento es el esperado.
- **El entorno de producción** es el entorno final en que una vez la aplicación ha sido extensamente validada en las fases anteriores, pasa a este entorno que es donde la aplicación se encuentra alojada para cumplir su función. Aquí no se realiza ninguna modificación en el código de la aplicación. A este entorno es al que tienen acceso los usuarios finales.

5.1 Pruebas unitarias y de integración

Durante el desarrollo de la aplicación se realizaron múltiples pruebas unitarias de integración, a continuación se mencionan algunas de ellas:

- En cuanto a las pruebas relativas a los usuarios se probó que los usuarios pudieran cambiar correctamente su contraseña desde la aplicación así como que los usuarios administradores pudieran cambiar su contraseña y la de otros usuarios. Además se probó que las modificaciones que pueden realizar los usuarios en las tablas fueras las permitidas, ya que muchos de los campos no deberían ser modificables o permitir ser eliminados por los usuarios.

- Las pruebas que requirieron de más tiempo fueron las relacionadas con la subida de ficheros y normalización de datos. Existían diversas casuísticas a la hora de subir los datos, dependiendo de la fuente de datos y de la información ya existente en la base de datos requería actualizar distintos campos, por lo que probar todos los casos fue una de las tareas más largas. Así como las comprobaciones de que las IP's contenidas en los scans de vulnerabilidades se encontraban previamente en los datos de la CMDB. En caso de que no se encontrasen, controlar todos los casos (IP's duplicadas, no existentes, pendientes de integración, servidor dado de baja…etc). Debido a que los volúmenes de datos a manejar eran muy grandes, estas pruebas se realizaron en todo momento con datos reales.
- Las pruebas relacionadas con la creación de reportes y dashboards fueron muy similares, requirieron de comprobaciones a la hora de realizar las consultas a la base de datos y de comprobar que los datos obtenidos eran los esperados. Se realizaron múltiples pruebas en las que los reportes se sacaban de forma manual y a través de la aplicación, para comprobar resultados.

5.2 Pruebas de usuario

Las pruebas de usuario se realizaron con más de 10 personas del departamento, los cuales validaron su funcionalidad en el entorno de pre-producción y posteriormente en el entorno de producción. Esta aplicación lleva más de 1 año en producción, permitiendo la gestión de vulnerabilidades en todo el departamento de forma unificada, por lo que durante los primeros meses errores o pequeñas mejoras detectadas por los usuarios fueron documentadas e implementadas en la aplicación. Debido a que es una aplicación que se usa diariamente por muchos usuarios, se puede decir que ha sido probada exhaustivamente y la versión actual de la misma es muy estable.

6 CONCLUSIONES Y TRABAJO FUTURO

El objetivo principal que se perseguía con este proyecto era una aplicación que facilitase la gestión de vulnerabilidades y generación de reportes de los clientes. En la aplicación desarrollada se cumplen todos los objetivos y requisitos mencionados a lo largo de esta memoria, ya que ésta permite unificar el resultado de distintas herramientas que realizan escáneres de vulnerabilidades, normalizar los datos y unificarlos con los contenidos en la CMDB, así como priorizar la gestión, dar métricas y permitir el seguimiento a los usuarios además de la generación de reportes para cada uno de los clientes.

En la siguiente tabla se muestran los tiempos mejorados en el proceso de gestión de vulnerabilidades:

Proceso	**Antes**	**Ahora**
Normalización de datos	8 h.	<1 min.
Personas trabajando simultáneamente en el mismo cliente	1 persona	todo el departamento
Generación de reportes	6 h.	< 1 min.
Guardar datos históricos por cliente	15 min.	< 3seg. (automático el día 1 de cada mes)
Cálculo del KPI	1 h por cliente	<5 seg para todos los clientes

Tabla 2. Comparativa de tiempos

Esta aplicación se ha convertido en la herramienta estándar del departamento para la gestión de vulnerabilidades. Se utiliza todos los días para dar seguimiento a todos los clientes que tienen contratado gestión de vulnerabilidades y ha supuesto al departamento una nueva forma de trabajar mucho más eficiente.

Este software está vivo, casi todos los meses se van incorporando pequeñas mejoras o funcionalidades que surgen de la evolución del proceso de gestión de vulnerabilidades a un grado de madurez mayor o de ideas nuevas que surgen en el departamento.

Actualmente en la siguiente lista se nombran los puntos de trabajo futuro en el desarrollo de la aplicación:

- Portal para el cliente. El reporte que se genera actualmente se entrega al cliente con la periodicidad pactada, pero se ha planteado la idea de crear un nuevo portal en el que cada cliente pueda ver en cualquier momento las vulnerabilidades de sus sistemas, así como el trabajo en progreso.
- Nuevas métricas y reportes. Se ha planteado la idea de diseñar nuevas métricas que puedan medir el trabajo realizado en el departamento y reportes ejecutivos de estos datos.
- Actualmente la aplicación realiza el cálculo del riesgo basándose en el estándar CVSS v2, los creadores del mismo han realizado una nueva versión (v3.), aunque actualmente no se emplea por la mayoría de los fabricantes se está pensando en poder integrarlo.

- Otro de los servicios que está cobrando fuerza en el departamento es la Gestión de *Advisories* de Seguridad (CERT), para este servicio no se dispone de ninguna aplicación y a día de hoy requiere de mucho trabajo manual que es posible automatizar. Este servicio está muy relacionado con la gestión de vulnerabilidades y se está trabajando en el análisis para su futura implementación.

7 BIBLIOGRAFÍA

[1] FIRST.org, Inc. *FIRST, Common Vulnerability Scoring System.* https://www.first.org/cvss último acceso: 10 de 05 de 2017).

[2] Forbes. https://www.forbes.com/sites/forbestechcouncil/2017/01/17/why-cybersecurity-should-be-the-biggest-concern-of-2017/2/#47879cc6274c (último acceso: 10 de 05 de 2017).

[3] «PWC.» https://www.pwc.es/es/publicaciones/gestion-empresarial/assets/temas-candentes-ciberseguridad.pdf (último acceso: 01 de 06 de 2017).

[4] Sans Institute. «Vulnerability Management: Tools, Challenges and Best Practices.» *Sans.* https://www.sans.org/reading-room/whitepapers/threats/vulnerability-management-tools-challenges-practices-1267 (último acceso: 12 de 05 de 2017).

[5] Tenable. «Nessus Tenable.» https://static.tenable.com/documentation/nessus_v2_file_format.pdf (último acceso: 21 de 05 de 2017).

A HERRAMIENTAS Y LENGUAJES

Para la implementación de esta aplicación se han utilizado las siguientes herramientas o ha sido necesario conocer los siguientes lenguajes de programación:

- Charts.js: librería que se emplea para diseñar las gráficas contenidas en los reportes y *dashboards*.
- Fullcalendar.io: se emplea para la realización del calendario, en el que se muestran todos los cambios en un formato visual acorde a la fecha y hora de realización.
- Datatables: Librería que se emplea para dar formato a la las tablas con paginación y búsquedas.
- WordPHP: Se emplea para la generación de reportes en formato Word.
- ExcelPHP: Se emplea para leer los ficheros de entrada en formato Excel a la aplicación como el de CMDB.
- Bootstrap: framework empleado para el diseño web de la aplicación.
- HTML: lenguaje empleado para la definición del contenido de cada página web.
- CSS: lenguaje empleado para dar estilo a las paginas web.
- PHP: lenguaje empleado para el desarrollo dinámico de la aplicación.
- JQuery: framework de Javascript que facilita la interacción con los elementos HTML, animaciones y AJAX de la páginas web.
- Javascript: lenguaje de programación que se emplea en páginas web para mejorar la interfaz de usuario y la dinámica del lado del cliente.
- MySQL: lenguaje empleado en la base de datos de la aplicación.
- AJAX: tecnología que se emplea para mantener una comunicación asíncrona en las páginas web.
- XML: lenguaje de etiquetas que permite la estructuración de un documento.
- JSON: formato de texto ligero para el intercambio de datos.

B ESTRUCTURA DE FICHEROS

En este anexo se describen los campos que contienen algunos que se pueden subir a la herramienta desarrollada.

B.1 Fichero de CMDB

En la CMDB se estructuran los datos teniendo en cuenta que cada cliente puede tener servicios, éstos a su vez pueden dividirse en entornos y cada entorno tiene sus servidores con sus respectivas IP's. Como ejemplo de estructura se puede ver en el siguiente diagrama:

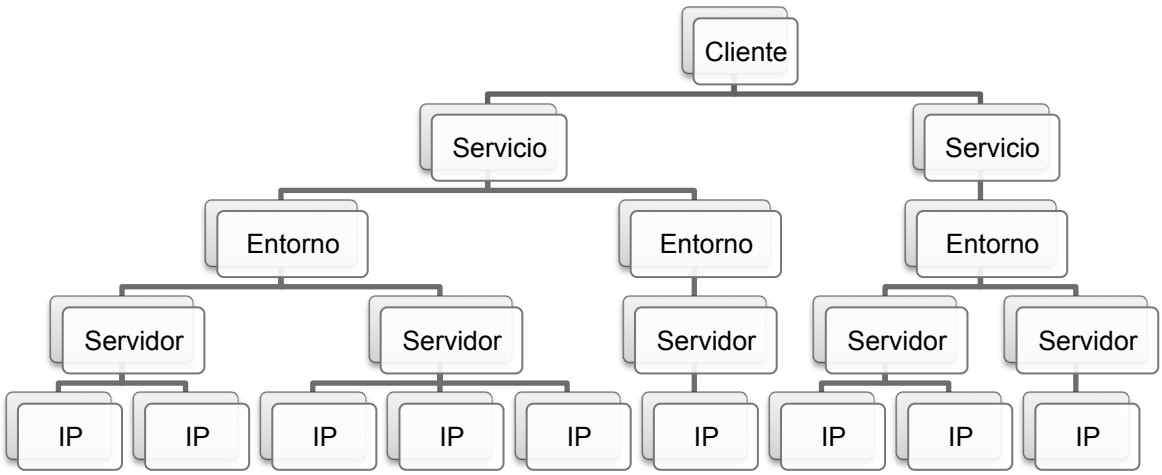

Figura 11. Ejemplo estructura de datos CMDB

El fichero de la CMDB ocupa entre 5 y 10 MB y cuando se sube a la aplicación éste se guarda a modo de histórico en el servidor en una ruta con la estructura cmdb/año/mes/día.

El fichero contiene una línea por cada IP de cada cliente. Los campos de este fichero que se emplean para poblar la base de datos son los siguientes:

- **Cliente:** nombre del cliente
- **Servicio:** nombre del servicio al que pertenece el servidor de esa línea del fichero.
- **SDM:** persona de contacto responsable del servicio
- **Entorno:** nombre del entorno al que pertenece el servidor de esa línea del fichero.
- **Tipo entorno:** tipos que puede tomar el entorno (producción, pre-producción, desarrollo, integración, otros).
- **Estado entorno:** estados que puede tomar el entorno (operativo, dado de baja, pendiente de integración, temporalmente parado).
- **Nombre servidor:** nombre que se le asigna al servidor.

- **Estado servidor:** estados que puede tomar el servidor (operativo, dado de baja, pendiente de integración, temporalmente parado).
- **Grupo propietario:** grupo dentro de la empresa responsable del servidor.
- **Funciones:** indica la funcionalidad general del servidor
- **Descripción:** campo adicional para ampliar la información del campo función
- **Sistema operativo:** versión de sistema de cada servidor.
- **IP:** una IP asociada al servidor
- **Función de red:** función de la IP
- **Fecha de instalación:** fecha en la que se monta el servidor
- **Centro:** centro donde se encuentra ubicado el servidor físico, en caso de los servidores virtuales indica la ubicación de la plataforma de virtualización (hipervisor).
- **Sala:** sala del centro en el que se encuentra el servidor físico.
- **Localización:** ubicación dentro de la sala.

B.2 Fichero de scan de vulnerabilidades Nessus

El formato de este fichero se encuentra definido en enlace mencionado en la bibliografía: (Tenable n.d.).