UNIVERSIDAD TECNOLÓGICA
DEL CENTRO DE VERACRUZ

Reporte Final de Estadía

Katia Villalbazo Rodríguez

Seguridad y herramientas para el desarrollo de aplicaciones SWF

Av. Universidad No. 350, Carretera Federal Cuitláhuac - La Tinaja
Congregación Dos Caminos, C.P. 94910. Cuitláhuac, Veracruz
Tel. 01 (278) 73 2 20 50
www.utcv.edu.mx

VERACRUZ Gobierno del Estado | SEV Estado de Veracruz | VER Educación Secretaría de Educación | SEP Secretaría de Educación Pública | DET Dirección de Educación Tecnológica del Estado de Veracruz

Universidad Tecnológica del Centro de Veracruz

Programa Educativo de Ingeniería en Tecnologías de la Información

Reporte para obtener su título de Ingeniero en Tecnologías de la Información

Proyecto de estadía realizado en Volkswagen de México S.A. de C.V.

Seguridad y herramientas para el desarrollo de aplicaciones SWF

Presenta: Katia Villalbazo Rodríguez

Cuitlahuac, Cuitlahuac. A 07 de Abril del 2017

Universidad Tecnológica del Centro de Veracruz

Programa Educativo de Ingeniería en Tecnologías de la Información

Nombre del Asesor Industrial: Lic. Demetrio Téllez Herrera

Nombre del Alumno: Katia Villalbazo Rodríguez

Resumen

El presente documento tiene como fin presentar el proceso de planeación y desarrollo de la elaboración de un proyecto. El proyecto consiste en dos mejoras en cuanto a la forma de trabajar del área de Software Factory y de la calidad que ofrecen con sus productos, esto mediante la implementación de mejoras en los procesos de ejecución que se toman en cuenta para llevar a cabo el desarrollo de un proyecto y con la propuesta de la utilización de una herramienta para la auditoria de seguridad de aplicaciones web, esto para buscar la calidad tanto del proceso de desarrollo como del producto final

El área de Software Factory actualmente presenta inconsistencia en cuanto a la documentación que entrega para la liberación de sus aplicaciones a productivo, lo que le exige buscar la manera de hacer este trabajo de forma más fácil y rápida. Por otro lado, la seguridad de las aplicaciones no se puede cubrir en cuento a todas las posibles vulnerabilidades porque se dispondría demasiado tiempo, que en esta área no se tiene, es así que se presenta la necesidad de encontrar una solución tecnológica que cubra con los aspectos necesarios de manera factible para presentar un análisis que apruebe la calidad de sus aplicaciones.

El desarrollo del mencionado proyecto se ha estimado como una investigación proyectiva para brindar la mejor solución a la problemática que se presenta, sostenida en un diseño de campo y documental. Asimismo para su modelado se ha empleado una metodología para el desarrollo de software de la que se basa el equipo de Software Factory, con el auxilio de implementar modelados gráficos para un mejor entendimiento de lo que se desea transmitir en esta solución.

Contenido

Ilustraciones

Introducción

Las tecnologías de la información y comunicación han destacado desde hace algunos años. Cada vez más esta rama se inserta y cobra fuerza en todos los sectores de las naciones. Su vertiginoso auge y su aplicación en innumerables áreas que han traído consigo resultados positivos cada vez mayores, constituyen un atractivo para todas las empresas que pretenden incrementar su desarrollo.

Ante este escenario Volkswagen de México ha optado por sistematizar algunos procesos que se llevan a cabo dentro de la empresa, que se encomiendan a un área específica llamada Software Factory (Fábrica de Software, SWF). El área se encarga de generar software desde cero, brindando soluciones a las demás áreas.

El trabajo que realiza Software Factory es de suma importancia dentro de la administración de la empresa, por lo que se debe asegurar un trabajo de calidad en cuanto a sus procesos y sus productos, es por ello que dentro de este documento se engloba la mejora de los procesos para el desarrollo de proyectos, tanto una mejora en cuanto a la detección de vulnerabilidades de seguridad de los productos (las aplicaciones web).

El área de la que se hace mención se ha percatado de que sus procesos no cuentan con la mejor calidad hasta el momento y que incluso excluyen algunos productos entregables que son de vital importancia para una buena documentación del proyecto, por lo que requieren una restructuración o mejora de dichos procesos. De igual manera, pretenden generar software de calidad para brindar un servicio de calidad y por eso la herramienta para la detección de vulnerabilidades de seguridad será de gran apoyo para cumplir su objetivo.

CAPÍTULO I: DATOS GENERALES

1. Historia

En 1954 una pequeña cantidad de empresarios unen sus esfuerzos para traer a México la sensación automovilística de Europa: El Sedán.

En los primeros días de ese año llegaron al puerto de Veracruz seis automóviles Sedán de lujo, dos Sedanes con techo corredizo, un Sedán convertible, una furgoneta tipo Panel, una ambulancia, un motor industrial y una chasis para demostración, cuyo destino era la Feria Exposición “Alemania y su Industria” en la capital del país. Los vehículos Volkswagen se presentan oficialmente en nuestro país, en la Ciudad Universitaria de México D.F.

Al iniciar 1954, el Príncipe Alfonso de Hohenlohe obtuvo la primera concesión para la distribución y servicio de automóviles de México, naciendo así la Distribuidora Volkswagen Central, S.A. La creciente demanda de los productos generó de inmediato la apertura de nuevos establecimientos en las ciudades de Puebla, Guadalajara y Monterrey.

Ilustración 1. Llegada de grupo Volkswagen a México

El convenio para ensamblar Sedanes en México se firma en Septiembre de 1954 con Fábricas Automex S.A. y se inicia con las primeras 250 unidades. Posteriormente, con la Studebaker-Packard de México, en junio de 1955, ser formaliza un acuerdo para continuar ensamblando las unidades tipo 113 hasta octubre de 1961.

En Xalostoc estado de México, los señores Ernesto Krause y Rómulo O'Farril compraron la firma Automóviles Ingleses S.A. para instalar la primera planta armadora PROMEXA (Promotora Mexicana de Automóviles), que inició actividades en Junio de 1962. Dos años después, en enero de 1964, cambió su denominación a Volkswagen de México S.A. de C.V.

Puebla reunió las condiciones óptimas para que VM edificara en sus inmediaciones el más importante proyecto automotriz a nivel nacional. Así, el 27 de febrero de 1965 se escritura la compra de 2 millones de metros cuadrados para la construcción de la nueva planta industrial, colocándose la primera piedra de la factoría y empezando de inmediato la obra, que en Julio de 1967 estaba lista para iniciar la producción.

La primera unidad Sedán producida en las nuevas y modernas instalaciones de la planta de Puebla salió de las líneas de producción el 23 de Octubre de 1967, sustituyendo de este modo la importación y el ensamble por la integración y la fabricación.

Ilustración 2. Primera producción del Sedán en Volkswagen de México

A partir de aquí Volkswagen de México ha emprendido muchos proyectos innovadores como aquel con el lema "Los llevamos sin rodar para que usted estrene". Que dio principio el 11 de diciembre de 1967 entregando unidades a los concesionarios de toda la República Mexicana a través de camiones nodriza.

El año 1968 significó grandes triunfos para la incipiente empresa, porque se alcanzaron dos importantes metas: la unidad de 100 000 producida en México y el primer lugar en ventas de automóviles.

Desde 1968 Volkswagen de México había puesto en marcha su programa de exportación de piezas y partes; fue en 1971 cuando logró exportar automóviles completamente terminados.

Para contar con instalaciones propias y adecuadas, el 18 de enero de 1973 fue inaugurado el Centro de Capacitación más moderno y equipado de la industria nacional, en un área de 2 727 metros cuadrados. En este mismo año con el suministro de 30 000 unidades tipo Safari enviadas en Marzo, Volkswagen de México inicia una nueva era participando en el mercado más exigente y competido del mundo es así que se comienza a exportar algunos sedanes a Estados Unidos.

En Abril de 1980 se inauguró la más moderna planta de fabricación de automóviles Volkswagen, con una capacidad de producción diaria de 1 200 motores enfriados por agua y 400 enfriados por aire. Así mismo Volkswagen celebró la producción del vehículo 1 000 000 para México en septiembre del mismo año, acontecimiento que significó un nuevo punto de partida para reafirmar la calidad, liderazgo empresarial y sólida estructura empresarial de VW.

Ilustración 3. Celebración de la producción del vehículo un millón en México

Los años siguientes fueron de crecimiento e impulso, que culminaron con la introducción de más nuevos modelos. Y reafirmaron la capacidad creativa del Consorcio Volkswagen, así como la excelencia generada por un complejo proceso de investigación y desarrollo automotriz, se presentó en 1984 el Corsar, cuya producción alcanzó las 32,966 unidades y el Corsar Variant en 1986.

Dos años más después llega el Pointer directo desde la fábrica Brasileña de Volkswagen, al mismo tiempo en que se introducía el Pointer ya empezaba la producción de Beetle en la planta de Puebla que actualmente es la única fábrica que produce el Beetle para todo el mundo. Pero fue hasta finales de 1996 que Volkswagen de México comenzó a hacer planes para producir el Beetle en México, en 1997 este ya se producía en serie en Puebla.

Ilustración 4. Producción del Beetle

En 1998 se anunciaba la llegada a nuestro país del nuevo Jetta, un auto tecnológicamente hablando muy avanzado, pero al igual que el Beetle este tardó en entrar a la producción en México, con equipo de lujo y paquetes muy completos este Jetta viene a imponer su presencia al igual que el nuevo Passat.

Volkswagen cierra 1998 alcanzando un volumen de producción cercano a 340 mil unidades.

En 1999 el Beetle es designado “El auto del año” en Norteamérica, el veredicto se presenta durante el Autoshow Internacional de Detroit. Este mismo mes se produce el

Beetle número 100 000. En febrero del 2000 se rompe record de ventas estimado del Jetta generación 4 en la planta armadora de Puebla de México.

En el mes de enero del 2003, Volkswagen AG dio a conocer que la planta de Volkswagen de México, ubicado en el estado de Puebla, había sido designada para iniciar en el 2004 la producción del nuevo Jetta A5 para todo el mundo.

En 2003, alrededor del 66% de la producción mexicana de Volkswagen se exportó a EE.UU. y Canadá, 12% a Europa y 1% a Sudamérica y el resto del mundo, quedando el 20% para el mercado nacional. Volkswagen de México inicia un proyecto que contempla la implantación de nuevas instalaciones de producción que afectan al taller de carrocería y a las líneas de montaje final para vehículos de tipo “Jetta A5” que cubrirán la capacidad utilizada por el antiguo “VW Beetle”. Además, las inversiones suplementarias en la nueva línea de producción de motores, el nuevo motor “R5 Super Ultra Low Vehicle Emission”, permitieran al Grupo VW ajustarse a las normas más estrictas sobre emisiones atmosféricas en Norteamérica.

Volkswagen de México celebró el 13 de agosto de 2013 la producción de 10 millones de automóviles y 11 millones de motores. La conmemoración fue enmarcada por la inauguración de un monumento en la planta de Puebla que simboliza la historia, tradición y evolución de Volkswagen en México.

Dicho monumento fue develado por el Gobernador del Estado de Puebla, Rafael Moreno Valle, acompañado por el Dr. Hubert Waltl, Miembro del Consejo Ejecutivo de Volkswagen, responsable para las Áreas de Producción y Logística, y el Sr. Andreas Hinrichs, Presidente del Consejo Ejecutivo de Volkswagen de México.

Ilustración 5. Inauguración del monumento del Vocho en la planta Puebla

En enero de 2014, Volkswagen de México celebró 50 años, y en ese marco se llevó a cabo el inicio de producción del Golf 7 en su planta de vehículos de Puebla.

2. Misión

Entusiasmar a nuestros clientes en todo el mundo con automóviles innovadores, confiables y amigables con el medio ambiente, así como con servicios de excelencia, para obtener resultados sobresalientes.

3. Visión

- Somos una empresa exitosa que genera utilidades de manera sustentable
- Somos líderes en el mercado mexicano, logrando satisfacer y retener al cliente ofreciendo un servicio excelente
- Somos competitivos y confiables en el desarrollo y la producción de vehículos y componentes
- Somos un socio comercial atractivo para proveedores y concesionarios, estableciendo con ellos relaciones sustentables

- Somos un equipo de colaboradores competentes, comprometidos y satisfechos
- Contamos con procesos innovadores, confiables y transparentes, enfocados a una calidad excelente y la satisfacción de nuestros clientes.

4. Principios

- Orientación a la mejora continua de nuestros procesos
- Cumplir con los requisitos nacionales, internacionales y del Grupo Volkswagen en materia de:
  - Calidad en los productos y servicios, prevención de la contaminación ambiental, seguridad y salud laboral
- Fomentar una actitud de excelencia en todos nuestros colaboradores y socios comerciales

5. Valores

- Cercanía al cliente
- Alto desempeño
- Crear valores
- Capacidad de renovación
- Respeto
- Responsabilidad
- Desarrollo sustentable

6. Organigrama

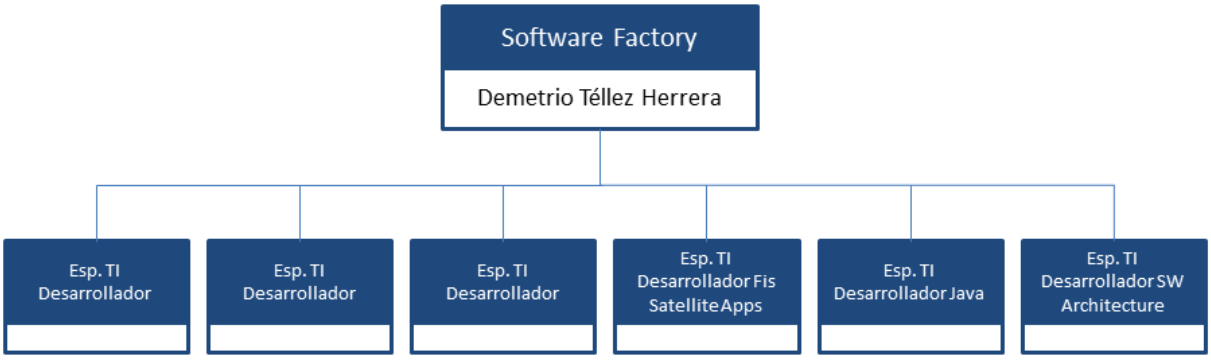

Ilustración 6. Organigrama de Volkswagen de México

CAPÍTULO II: ANÁLISIS DEL PROBLEMA Y PROPUESTA DE LA SOLUCIÓN

7. Planteamiento del problema

En el área de Software Factory de Volkswagen de México se han presentado algunos problemas que generan complicaciones en cuanto al tiempo que requiere liberar una aplicación a productivo, esto debido a que el área que se hace responsable del mantenimiento de las aplicaciones ya liberadas (AMS no SAP) exige una cantidad de documentos que Software Factory comienza a realizar hasta el final del ciclo de vida del proyecto y que cada integrante realiza de acuerdo a su criterio. Por esta razón en ocasiones los documentos son rechazados por AMS no SAP, lo que hace que la liberación se atrase y reduce la calidad que tienen como equipo porque no se respetan los tiempos establecidos.

Los procesos que se realizan hasta el momento dentro del área para cada proyecto, es decir, de acuerdo a la metodología que adoptan, no cubren con todas las actividades y productos entregables necesarios para ser entregados al área de AMS no SAP antes mencionada. Esta área hace entrega de un checklist de los documentos específicos que deben ser entregados para el trámite de posesión de la aplicación. Es por ello que este proceso es muy importante y hasta el momento las actividades no abarcan los documentos necesarios o no se tiene un acuerdo para trabajar en un formato específico, lo que genera malas prácticas y mala organización.

Por lo general las aplicaciones que se desarrollan son en un entorno web, lo que las hace más vulnerables en cuento a ataques maliciosos de terceras personas. Para solucionar estos problemas de calidad, en la actualidad el área de Software Factory hace pruebas a detalle para verificar que las aplicaciones cuenten con un grado menor de vulnerabilidad, de acuerdo a lo que ellos puedan cubrir. Pero esto les lleva más tiempo de lo que desearían y no logran cumplir con todas las pruebas, quedando la incertidumbre de si se entregan con vulnerabilidades.

Lo antes mencionado trae consigo más problemas, debido que al contar con información de suma importancia que administran las aplicaciones, la seguridad debe ser uno de los atributos más confortables que se puede ofrecer. Además de que las aplicaciones después de ser entregadas al departamento del que fue la petición, pasan a ser parte del departamento de AMS non SAP. Siendo así este departamento exige que las aplicaciones sean entregadas con la mejor calidad posible, pero también se exige la entrega a tiempo de cada proyecto, lo que genera algunos desacuerdos entre las áreas involucradas.

Otro punto importante es que no se hace una planeación previa de las vulnerabilidades que se deben evitar y algunos desarrolladores las hacen omisas siendo hasta el final del desarrollo cuando se tratan de solucionar. Esto es una mala organización que el equipo ha ido arrastrando desde proyectos pasados donde la mejor opción sería hacer un análisis y prever estos acontecimientos.

8. Propuesta de solución

Para la solución de los problemas antes mencionados se pretende evaluar herramientas para la auditoria y escáner de aplicaciones web, las cuales ofrecen un análisis detallado de las vulnerabilidades en cuanto a seguridad, estas aplicaciones detectan si la aplicación es vulnerable a inyecciones SQL, ataques XSS, inclusión de archivos, entre otras. Esto con el fin de encontrar una herramienta que sea adecuada y funcione de la mejor manera para que se implemente en la creación de proyectos del área de Software Factory y permita evaluar las aplicaciones de acuerdo a una métrica establecida.

Esto permite ahorrar el tiempo que se emplea actualmente, debido a que la herramienta detectará las complicaciones y el equipo de desarrollo ya solo se dedicará

a emplear las mejoras. De igual manera las aplicaciones no serán liberadas al menos de que cubran con la métrica establecida.

En cuanto a la mejora de procesos se evaluará a detalle cada fase de la metodología y como se está llevando a cabo hoy en día, para hacer las modificaciones de acuerdo al seguimiento de todos los procesos que aún no se han implementado o que se llevan de manera errónea y así mismo incluir cada producto faltante en la fase que más se acople, con el fin de que al momento de la entrega al área de AMS non SAP algunos productos entregables ya estén listos para ser liberados y así no tome más tiempo el concluir esta fase.

9. Objetivos

9.1. Objetivo General

Emplear mejoras en los procesos de la metodología para el desarrollo de software y evaluar herramientas para auditar y escanear la seguridad web de las aplicaciones que se desarrollan en el área de Software Factory de Volkswagen de México.

9.2. Objetivos Específicos

- Investigar acerca de herramientas para la auditoria y escáner de aplicaciones web para comparar sus ventajas y desventajas.
- Comparar el funcionamiento de cada herramienta a evaluar para elegir la que mejor se adapte a los requerimientos necesarios para el área de Software Factory.
- Verificar el funcionamiento de la herramienta, evaluando cada una de las funciones disponibles y módulos a los que se tienen acceso.
- Elaborar pruebas conjuntamente con el personal del área de Software Factory para verificar que cumpla con las expectativas deseadas.

- Implementar una métrica que establezca el margen de vulnerabilidad que puede ser permitido por cada aplicación.
- Implantar la aplicación para que pueda ser utilizada, realizando la respectiva capacitación de los usuarios finales.
- Colocar los productos entregables en la fase correspondiente para mejorar el proceso de desarrollo.
- Actualizar los diagramas de comportamiento que se generan para dar explicación grafica de como fluye los procesos y los productos entregables.
- Actualizar los formatos de plantillas para cada producto entregable o crearlo si aún no se ha implementado.
- Actualizar el repositorio, ordenando cada fase con sus respectivos productos entregables.

10. Justificación

10.1. Justificación

Los motivos por los que se pretende desarrollar el proyecto son porque se quiere contar con una herramienta que ayude a validar la calidad de las aplicaciones que se desarrollan en el área de Software Factory, haciendo más fiable el trabajo que realizan los desarrolladores, para que al momento de hacer entrega del proyecto a el área de AMS non SAP se pueda validar que se cumple con la calidad deseada.

El tiempo que se requiere para identificar las posibles vulnerabilidades dentro de las aplicaciones web es muy considerable a comparación del proceso que en la actualidad se lleva a cabo y el resultado es más exacto y fiable. Evaluar una aplicación de acuerdo a la calidad de seguridad con la que cuenta llevará solo unos minutos o en todo caso dependerá de la robustez de esta, pero siempre respetando un tiempo

mínimo para visualizar resultados. El sistema se ajustará a las necesidades del equipo de desarrollo para ahorrar tiempos innecesarios.

Utilizar una herramienta que permite evaluar las aplicaciones web para encontrar vulnerabilidades de seguridad podrá asegurar una forma de trabajo más confiable y ayudará a mejorar las prácticas de los desarrolladores en cuanto a al código.

Mejorar los procesos de la metodología que actualmente se implementa, ayudará a que se lleve a cabo un trabajo de calidad y que todo se realice en el tiempo en que se tiene planeado, que los productos entregables puedan transmitir lo que realmente se solicita y que las áreas involucradas puedan trabajar de manera más organizada.

10.2. Beneficios y beneficiados

Se agilizarán los tiempos, permitiendo dedicarle menos tiempo a la actividad de verificar la calidad de las aplicaciones y con ello el proyecto podrá contar con un porcentaje mayor a ser entregada en tiempo y forma.

Los principales beneficiados con la implementación de la herramienta de evaluación de seguridad será el equipo de desarrollo del área de Software Factory dado a que facilitará la forma en que se llevan a cabo las pruebas hasta el momento, ahorrando tiempo y recursos. El área de AMS non SAP también se beneficiará con respecto a la validación que realizan, haciendo más fácil su trabajo sin tener que dedicar demasiado tiempo a ese aspecto.

El cliente o el departamento que ha solicitado la aplicación tendrán una mayor seguridad de que la información administrada por la aplicación estará bien manejada y no será obstruida por terceras personas. La empresa en general será beneficiada por esta herramienta.

11. Viabilidad del proyecto

El proyecto demuestra ser viable en cuestiones técnicas, sociales y económicas.

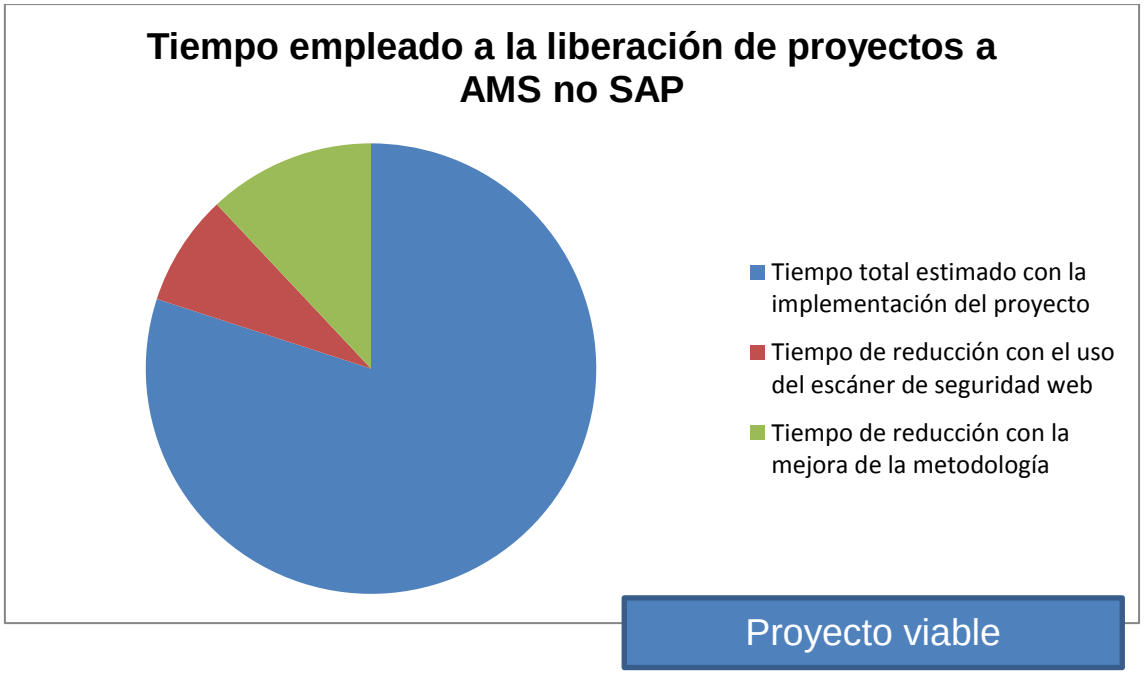

Ilustración 7. Demostración grafica de la viabilidad del proyecto

11.1. Viabilidad Técnica

De acuerdo al estudio de viabilidad técnica con respecto a la herramienta de auditoria de seguridad para aplicaciones web, el proyecto es viable para su implementación en el área de Software Factory de Volkswagen de México. El proyecto prácticamente complementa el trabajo que realiza el área, que se dedica a crear productos software para las demás áreas que lo necesiten dentro de la empresa.

El implementar el proyecto ahorra tiempos en la elaboración de los proyectos que ellos desarrollan y en la liberación al área de AMS no SAP, que es donde el equipo de Software Factory pierde relación con el sistema realizado.

La aplicación que se va a utilizar es una herramienta conocida que se ajustará a las necesidades actuales del equipo de trabajo, que sea fácil de manejar y funcione por

encima de otras aplicaciones del mismo mercado, brindando el menor margen de error posible.

La herramienta deberá ser portable. Hasta el momento existen diversas aplicaciones de este tipo que funcionan en los principales sistemas operativos, sean Windows, Linux y Mac OS. Esto hará más fácil la elección de una de ellas y que también permita mostrar una interfaz gráfica para mayor usabilidad hacia el usuario final.

La persona que evaluará las aplicaciones y así mismo la aplicación que se va a ejecutar por elección, cumple con los requisitos para hacer pruebas y evaluar el funcionamiento de todas las posibles opciones, lo cual hace que el proyecto sea viable en cuanto al conocimiento que se cuenta para dichas tareas.

Se tiene el equipo de cómputo necesario para los requisitos de las aplicaciones, tanto para hacer las pruebas como para la implementación con cada integrante del equipo.

En el caso de la mejora en el proceso de la metodología, se cuentan con los recursos, el personal y los conocimientos que requiere este cambio en el proceso. Todas las áreas involucradas participarán de forma amena para hacer un mejor trabajo dentro del área específica.

11.2. Viabilidad económica

Económicamente el proyecto ha demostrado ser rentable, en dado caso los recursos monetarios no son un requisito fundamental para llevar a cabo el proyecto porque se pretende implementar una aplicación de libre distribución y el proyecto es realizado en prácticas profesionales, lo cual no implica aspectos económicos respecto a la persona que realizará el trabajo.

Debido a que no se ha autorizado un presupuesto para adquirir una aplicación con licencia, es que se busca implementar una que sea competitiva y al mismo tiempo de distribución libre. Los demás gastos implicados con respecto a los servicios e instalaciones son responsabilidad de planta, como a continuación se menciona.

- Se hará uso de herramientas tecnológicas gratuitas, para no hacer una inversión en licencias de ningún tipo.
- Para el desarrollo se ocuparán los equipos de cómputo con los que ya cuenta la empresa, asignado uno a cada usuario.
- El lugar donde se desarrollará el proyecto será en las mismas instalaciones de la empresa.
- El pago de servicios será responsabilidad de la empresa, con respecto a la energía eléctrica que requieren los equipos de cómputo.

11.3. Viabilidad Social

El proyecto es viable socialmente porque con su desarrollo habrá muchas personas involucradas y favorecidas, a continuación se hace mención de los involucrados:

- Volkswagen de México
- Equipo Software Factory
- Equipo AMS no SAP
- Área organización de TI

El área de Software Factory podrá mejorar los procesos que realiza al momento de entregar los sistemas a mantenimiento (AMS no SAP) y con ello hacer más rápido el proceso, lo que será realmente considerable para los usuarios involucrados del futuro sistema. Los sistemas podrán incorporarse de manera más sencilla y así poder operarlos lo más pronto posible.

La(s) personas que desarrollarán el proyecto conocerán más acerca de las necesidades que se presentan en el ámbito laboral y podrán tener una interacción más a fondo con las herramientas básicas que solicita un equipo de desarrollo de software. Al cumplir satisfactoria con la finalización del proyecto se cubrirá el rubro para la titulación de Ingeniería en Tecnologías de la Información.

12. Alcances

Implementar una herramienta que evalué la seguridad de las aplicaciones que se desarrollan en Software Factory y que se adapte a las necesidades actuales de cada proyecto, lo que implica:

- Que sea capaz de hacer un escáner detallado de las aplicaciones para mostrar sus posibles vulnerabilidades.
- Que se pueda adaptar a las vulnerabilidades específicas de acuerdo al entorno de desarrollo utilizado.
- Generar métricas las cuales deban cumplir todas las aplicaciones para asegurar una buena calidad.
- Que se capacite a los desarrolladores ara que puedan hacer el mejor uso de la herramienta empleada.

Mejorar los procesos que se llevan a cabo hasta el momento de acuerdo al desarrollo de proyectos y la metodología implementada, para adaptar la correcta entrega y creación de los productos que se generan, lo que implica:

- Adecuar los procesos de la metodología de manera más factible de acuerdo a las buenas prácticas.
- Adecuar y crear las plantillas de los productos entregables en la fase adecuada para reducir tiempo y contar con una mejor organización.

- Actualizar el documento base de los procesos, restableciendo los diagramas de flujo de información y añadiendo los puntos importantes que se han omitido.

Todos los procesos de la herramienta, tanto como el de la metodología que se mencionan anteriormente serán desarrollados en un plazo máximo de 4 meses.

La herramienta será implementada para el uso exclusivo del área de Software Factory, con única capacitación para dicha área.

13. Limitaciones

- El proyecto se realizará en un tiempo máximo de 4 meses

  El proyecto está planeado para un tiempo máximo de cuatro meses, es por ello que toda la planeación se integró para respetar el periodo definido. Es importante mencionar que se cumplirán con los requerimientos planteados hasta el momento, pero en caso de solicitar algunas mejoras y el tiempo ya haya finalizado, se hará omisa la petición.

- No se mejorarán los problemas de seguridad que se detecten con la nueva herramienta

  Se harán las pruebas pertinentes con la herramienta de auditoria de seguridad para verificar su funcionamiento y obtener métricas de vulnerabilidad disponible, más no se mejorarán los problemas de seguridad que estás presenten.

- Solo se dará capacitación del uso de la herramienta de auditoria de seguridad en un entorno de Windows 7

  La herramienta se va a implementar en el sistema operativo Windows que es el SO con el que trabaja Volkswagen de México, es por esto que la capacitación hacia

los desarrolladores solo se implementará en un ambiente del SO antes mencionado.

- Se brindará una introducción de los entregables del modelo CMMI, más no se introducirá a su implementación.

Al finalizar con la mejora de los nuevos procesos dentro de la metodología que actualmente se utiliza, se hará una comparación con los procesos que contempla CMMI dando una introducción a los que hace falta para logar la certificación en dicho modelo. La implementación del modelo ya no es parte del objetivo de este proyecto.

CAPÍTULO III: FUNDAMENTACIÓN TEÓRICA

14. Marco Teórico

La razón de ser de este documento tiene como principal enfoque dos actividades a realizar, una de ellas es el cómo hacer uso de una tecnología para la auditoria y escáner de aplicaciones web, donde para comenzar hay que conocer muy a detalle que es una aplicación web y que posibles vulnerabilidades de seguridad pueden presentar como para buscar implementar una de estas herramientas. La otra actividad consta de la actualización de los procesos que realiza el equipo de desarrollo de Software Factory, la organización al momento de entregar los productos generados y demás mejoras que son necesarias hasta el momento, teniendo como base que trabajan con una metodología ágil llamada SCRUM.

Para comenzar con este marco teórico es necesario saber acerca de que son las aplicaciones web. Las aplicaciones web reciben este nombre porque se ejecutan en la internet. Es decir que los datos o los archivos en los que trabajas son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en tu computador.

El concepto de aplicaciones web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y nos envían a nuestros dispositivos o equipos los datos que requerimos en ese momento, quedando una copia temporal dentro de nuestro equipo. *En cualquier momento, lugar y desde cualquier dispositivo podemos acceder a este servicio, sólo necesitamos una conexión a internet y nuestros datos de acceso, que por lo general son el nombre de usuario y contraseña.*

En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como

JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (renderice) la aplicación.

Una de las ventajas de las aplicaciones web cargadas desde internet (u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. También la posibilidad de ser ejecutadas en múltiples plataformas por la fácil portabilidad de estas aplicaciones en los navegadores web.

Las aplicaciones web al ser sistemas ejecutados desde un navegador y al salir a un entorno de internet, son más susceptibles a vulnerabilidades en cuanto a ataques de terceros o ciberdelincuentes, que buscan solo aprovechar esas vulnerabilidades para sacar algún provecho o solo causar un daño.

Se debe entender que programar aplicaciones web seguras no es una tarea fácil, ya que requiere por parte del programador, no sólo cumplir con el objetivo funcional básico de la aplicación, sino una concepción general de los riesgos que puede correr la información procesada por el sistema. Gran parte de los problemas de seguridad en las aplicaciones web son causados por la falta de seguimiento en dos rubros muy importantes de los que depende cualquier aplicación, las entradas y salidas del sistema.

Ilustración 8. Entradas y salidas de aplicaciones web

Es conveniente emplear medidas de seguridad que sean transparentes a los usuarios y que no resulten engorrosas en su empleo. Por ejemplo, el uso de un acceso que solicita el nombre de usuario y contraseña, permite controlar el acceso de los usuarios hacia secciones restringidas de la aplicación. Este paso adicional, es una característica que impacta en la rapidez de acceso a la información por parte del usuario, pero que proporciona un elemento adicional de protección.

A mayor complejidad de nuestro sitio, aumenta el riesgo de que se sufra un ataque debido a sus características más elaboradas, es por eso que deben considerarse opciones de seguridad necesarias y sencillas pero eficientes, que ayuden a mitigar cualquier característica que la haga vulnerable.

En la actualidad existe un sinfín de ataques hacia las aplicaciones web, los cuales las vuelven vulnerables en cuanto a las prácticas maliciosas que se puedan generar de acuerdo a los acontecimientos, pero antes de adentrar al tema de los distintos ataques, es importante saber otros aspectos que engloban las aplicaciones web.

Lo principal para el desarrollo de una aplicación web es saber y evaluar que lenguaje de programación se va a utilizar. Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a una computadora.

Existen varios cientos de lenguajes y dialectos de programación diferentes. Algunos se crean para una aplicación especial, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones. En todo caso los lenguajes de programación deben tener instrucciones que pertenecen a las categorías ya familiares de entrada/salida, cálculo / manipulación de textos, lógica / comparación y almacenamiento/recuperación. Según Carlos Ureña un lenguaje de programación es "*Conjunto de reglas o normas que permiten asociar a cada programa*

correcto un cálculo que sería llevado a cabo por un ordenador (sin ambigüedades)". Dado a estos conceptos concluimos que un lenguaje de programación tienen un conjunto de instrucciones que permiten realizar dichas operaciones. Existe una marcada diferencia en los símbolos, caracteres y sintaxis de los lenguajes de máquina, lenguajes ensambladores y lenguajes de alto nivel.

Ilustración 9. Lenguajes de programación

Cabe mencionar que una computadora es totalmente inútil si no dispone de un programa capaz de procesar información, para que se realice dicho procesamiento de información habrá sido necesario pensar, construir, y crear un programa y ejecutar dicho programa o aplicación en la computadora. Existen diversos lenguajes de programación ya establecidos, de los cuales les haré mención a continuación: C#, Delphi, C, PHP, Perl, Python, Visual Basic, Pascal, Java, entre otros. Más adelante se hará mención de alguno de estos lenguajes de programación.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas.

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

La mayoría de los lenguajes de programación se complemnetan con Html para la creación de aplicaciones web. Uno de los lenguajes más utilizados es Javascript, este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

El código Javascript puede ser integrado dentro de nuestras páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseño un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).

Otro de los lenguajes más utilizados es PHP, que es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Y por último es importante mencionar el lenguaje JSP. Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat. Sus características son que el código se encuentra separado de la lógica del programa, las páginas son compiladas en ela primera petición, permite sepearra la parte dinámica con la estática en las páginas web y que los archivos se encuentran con la extención (jsp).

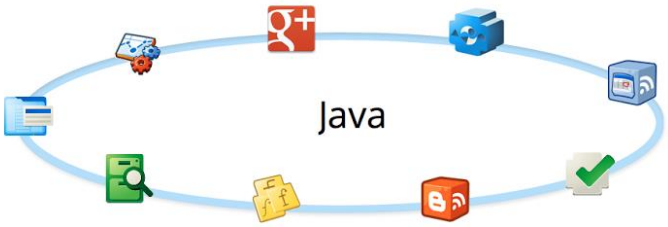

Ilustración 10. Sistemas o aplicaciones desarrolladas con Java

Para que el sistema funcione a la perfección y se pueda mantener la integridad de los datos es necesario contar con un lugar, por llamarlo así, que nos permita almacenar cada uno de estos datos, asegurando su uso adecuado. Para esto se utilizan las bases de datos, que son es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.

Para la administración de las bases de datos es necesario contar con un gestor que nos permita controlar cada uno de los procesos y consultas que se puedan realizar

para el sistema. Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server, etc.

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Ilustración 11. Sistemas Gestores de Base de Datos

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web

dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes.

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras.

Ahora ya es momento de abordar lo diferentes tipos de ataque que existen y cuáles son los más populares y peligrosos. Una de las vulnerabilidades más conocidas en la inyección SQL. Un ataque por inyección SQL consiste en la inserción o “inyección” de una consulta SQL por medio de los datos de entrada desde el cliente hacia la aplicación. Un ataque por inyección SQL exitoso puede leer información sensible desde la base de datos, modificar la información (Insert/ Update/ Delete), ejecutar operaciones de administración sobre la base de datos (tal como parar la base de datos), recuperar el contenido de un determinado archivo presente sobre el sistema de archivos del DBMS y en algunos casos emitir comandos al sistema operativo. Los ataques por inyección SQL son un tipo de ataque de inyección, en el cual los comandos SQL son insertados en la entrada de datos con la finalidad de efectuar la ejecución de comandos SQL predefinidos. (Romaniz, 2015)

Una inyección SQL ocurre cuando:

- Los datos ingresan en un programa desde una fuente que no es de confianza.
- Los datos construyen dinámicamente una consulta SQL.

Las principales consecuencias son:

Confidencialidad: Dado que las bases de datos SQL generalmente almacenan información sensible, la pérdida de la confiabilidad es un problema frecuente con las vulnerabilidades de inyección SQL.

Autenticación: Si se utilizan consultas SQL pobres para chequear nombres de usuarios u contraseñas, puede ser posible conectarse a un sistema como otro usuario sin conocimiento previo de la contraseña.

Autorización: Si la información de autorización es almacenada en una base de datos SQL, puede ser posible cambiar esta información mediante la explotación exitosa de una vulnerabilidad por inyección SQL.

Integridad: Así como puede ser posible leer información sensible, también es posible realizar cambios o incluso borrar esta información mediante un ataque por inyección SQL.

Otro de los ataques más conocidos en este ámbito es el ataque XSS (Cross-Site Scripting) que es un ataque de inyección de código malicioso para su posterior ejecución que puede realizarse a sitios web, aplicaciones locales e incluso al propio navegador.

Sucede cuando un usuario mal intencionado envía código malicioso a la aplicación web y se coloca en forma de un hipervínculo para conducir al usuario a otro sitio web, mensajería instantánea o un correo electrónico. Así mismo, puede provocar una negación de servicio (DDos).

Generalmente, si el código malicioso se encuentra en forma de hipervínculo es codificado en HEX (basado en el sistema de numeración hexadecimal, base 16) o algún otro, así cuando el usuario lo vea, no le parecerá sospechoso. De esta manera,

los datos ingresados por el usuario son enviados a otro sitio, cuya pantalla es muy similar al sitio web original.

De esta manera, es posible secuestrar una sesión, robar cookies y cambiar la configuración de una cuenta de usuario.

También existe la inclusión de archivos que consiste en una vulnerabilidad existente solamente en páginas dinámicas en PHP que permite el enlace de archivos remotos situados en otros servidores a causa de una mala programación de la página que contiene la función include().

Es verdad que existen muchos tipos de ataques que pueden ser de alta gravedad, dado a que cada uno de ellos tiene la forma de afectar al servidor, al usuario o al propietario de la aplicación.

La metodología Scrum es un proceso de desarrollo de software iterativo y creciente utilizado comúnmente en entornos basados en el desarrollo ágil de software. Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas.

Valores

- Adaptabilidad a los cambios entre iteraciones
- Blindaje de cada iteración con respecto al cambio
- Pequeña jerarquía definida
- Stakeholders->Product Owner->Scrum Master->Equipo

Conclusiones

- Actuación por sentido común
- Sencillo de entender

- Auto-organización del equipo
- Rápido sin necesidad de planificaciones iniciales como Pert o diagrama de Gantt

La metodología XP (eXtreme Programming) es la más destacada de las metodologías ágiles del desarrollo del software. Fue ideada por Kent Beck a finales de los 90. En XP los cambios en los requisitos son un aspecto natural del desarrollo de los proyectos, por tanto: Adaptabilidad>Previsión en el diseño del software.

Principios básicos

- Testeos continuos
- Planificación
- Pequeñas mejoras (frecuentes entregas)
- Sistema de metáforas (nombres claros)
- Diseño simple (rápido, funciones necesarias)
- Refactorización del código
- Programación por parejas
- El código es de todos
- Programación estandarizada / simple
- Ritmo sostenible
- Relación con el cliente

Comparativa SCRUM – XP

Ambas son metodologías de desarrollo ágiles, basadas en los valores del “agile manifiesto”. El hecho de que en ambas se utilicen las historias de usuarios. Que se realicen continuamente entregas al cliente en cortos períodos de tiempo. Las reuniones exprés, de pie, entre los miembros del equipo.

SCRUM	EXTREME PROGRAMMING (XP)
El Scrum Team trata de seguir el orden de prioridad que marca el Product Owner en el Sprint Backlog pero si ven que es mejor modificar el orden de prioridad para el desarrollo de las tareas, pueden hacerlo.	El equipo de desarrollo sigue estrictamente el orden de prioridad de las tareas definidas por el cliente (aunque el equipo de desarrollo le ayude a decidir, ellos son los que mandan)
El Scrum es una metodología agil más basada en la administración del proyecto.	En cambio, el XP se centra en la propia programación o creación del producto.
Los itinerarios de entrega son de dos a cuatro semanas y se conocen como sprint.	Las iteraciones de entrega son de una a tres semanas (algo más rápidas)
Al finalizar un sprint, las tareas que se han realizado del Sprint Backlog y en las que el Product Owner ha mostrado su conformidad ya no se vuelve a tocar en ningún momento. “Lo que se termina funciona y está bien, se aparta y ya no se toca”	Las tareas que se van terminando en las diferentes entregas al cliente son susceptibles a modificaciones duran6te el transcurso de todo el proyecto, incluso después de que funcione correctamente.

Ilustración 12. Tabla comparativa entre SCRUM y XP

15. Metodología de desarrollo

Dentro de los muchos frameworks para desarrollar software, Scrum se ha convertido en los últimos tiempos en un estándar que muchas grandes empresas de tecnología están utilizando para acortar sus tiempos de desarrollo, y entregar un producto de calidad.

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Ilustración 13. Flujo de un sprint

El Sprint es el ritmo de los ciclos de Scrum. Está delimitado por la reunión de planificación del sprint y la reunión retrospectiva. Una vez que se fija la duración del sprint es inamovible. La mayoría de los equipos eligen dos, tres o cuatro semanas de

duración. Diariamente durante el sprint, el equipo realiza una reunión de seguimiento muy breve. Al final del sprint se entrega el producto al cliente en el que se incluye un incremento de la funcionalidad que tenía al inicio del sprint.

El proceso parte de la lista de requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente ha priorizado los requisitos balanceando el valor que le aportan respecto a su coste y han sido divididos en iteraciones y entregas. (Gallego, 2013)

15.1. Actividades de la metodología Scrum

Las actividades que se plantea realizar en la metodología Scrum son las siguientes:

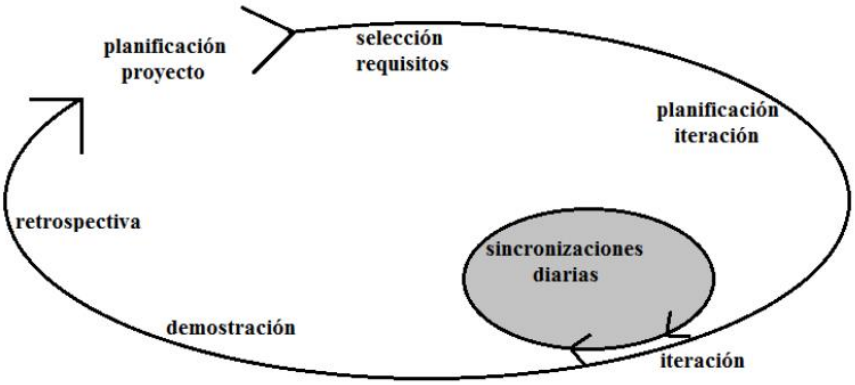

Ilustración 14. Actividades implementadas en Scrum

Planificación de la iteración

La planificación de las tareas a realizar en la iteración se divide en dos partes:

Primera parte de la reunión. Se realiza en un tiempo máximo 4 horas:

- El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto, pone nombre a la meta de la iteración (de manera que ayude a tomar

decisiones durante su ejecución) y propone los requisitos más prioritarios a desarrollar en ella.

- El equipo examina la lista, pregunta al cliente las dudas que le surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

Segunda parte de la reunión. Se realiza en un tiempo máximo 4 horas. El equipo planifica la iteración, dado que ha adquirido un compromiso, es el responsable de organizar su trabajo y es quien mejor conoce cómo realizarlo.

- Define las tareas necesarias para poder completar cada requisito, creando la lista de tareas de la iteración.
- Realiza una estimación conjunta del esfuerzo necesario para realizar cada tarea.
- Cada miembro del equipo se asigna a las tareas que puede realizar.

Ejecución de la iteración (sprint)

- En Scrum un proyecto se ejecuta en iteraciones de un mes natural (pueden ser de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite.
- Cada día el equipo realiza una reunión de sincronización, donde cada miembro inspecciona el trabajo de los otros para poder hacer las adaptaciones necesarias, así cómo comunicar cuales son los impedimentos con que se encuentra.
- El Facilitador (Scrum Master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad. Elimina los obstáculos que el equipo no puede resolver por sí mismo. Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

Reunión diaria de sincronización del equipo (Scrum daily meeting)

El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad.

Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para al finalizar la reunión poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso conjunto que el equipo adquirió para la iteración (en la reunión de planificación de la iteración).

Cada miembro del equipo debe responder las siguientes preguntas en un intervalo de tiempo de cómo máximo 15 minutos:

- ¿Qué he hecho desde la última reunión de sincronización? ¿Pude hacer todo lo que tenía planeado? ¿Cuál fue el problema?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener para cumplir mis compromisos en esta iteración y en el proyecto?

Como apoyo a la reunión, el equipo cuenta con la lista de tareas de la iteración, donde se actualiza el estado y el esfuerzo pendiente para cada tarea, así como con el gráfico de horas pendientes en la iteración.

Se actualiza la gráfica burndown con el trabajo realizado.

Demostración de requisitos completados (Sprint Demonstration)

- Reunión informal donde el equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado

para ser entregado con el mínimo esfuerzo, haciendo un recorrido por ellos lo más real y cercano posible al objetivo que se pretende cubrir.

- En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
- Se realiza en un tiempo máximo 4 horas.

Retrospectiva (Sprint Retrospective)

El equipo analiza cómo ha sido su manera de trabajar durante la iteración, qué cosas han funcionado bien, cuáles hay que mejorar, qué cosas quiere probar hacer en la siguiente iteración, qué se ha aprendido y cuáles son los problemas que podrían impedirle progresar adecuadamente, con el objetivo de mejorar de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados que el propio equipo no pueda resolver por sí mismo.

Se realiza en un tiempo máximo 3 horas.

Replanificación del proyecto

Durante el transcurso de una iteración, el cliente va trabajando en la lista de requisitos priorizada del producto o proyecto, añadiendo requisitos, modificándolos, eliminándolos, repriorizándolos, cambiando el contenido de iteraciones y definiendo un calendario de entregas que se ajuste mejor a sus nuevas necesidades.

Los cambios en la lista de requisitos pueden ser debidos a:

- Modificaciones que el cliente solicita tras la demostración que el equipo realiza al final de cada iteración sobre los resultados obtenidos, ahora que el cliente entiende mejor el producto o proyecto.
- Cambios en el contexto del proyecto (sacar al mercado un producto antes que su competidor, hacer frente a urgencias o nuevas peticiones de clientes, etc.).

- Nuevos requisitos o tareas como resultado de nuevos riesgos en el proyecto.

15.2. Roles

Cuando se aplica la metodología Scrum se determinan las responsabilidades siguientes:

No hay un jefe de proyecto. Las responsabilidades del tradicional jefe de proyecto se distribuyen a los siguientes roles de un equipo Scrum:

- El cliente o Product Owner

- Scrum master o facilitador

- Resto del equipo

Cliente (Product Owner)

Las responsabilidades del Cliente (que puede ser interno o externo a la organización) son:

- Ser el representante de todas las personas interesadas en los resultados del proyecto (internas o externas a la organización, promotores del proyecto y usuarios finales) y actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones.
- Definir los objetivos del producto o proyecto.
- Dirigir los resultados del proyecto.
- Es el propietario de la planificación del proyecto: crea y mantiene la lista priorizada con los requisitos necesarios para cubrir los objetivos del producto o proyecto, conoce el valor que aportará cada.
- Divide la lista de requisitos estableciendo un calendario de entregas.

- Participar en la reunión de demostración de la iteración, revisando los requisitos completados.

Facilitador (Scrum Master)

Lidera al equipo llevando a cabo las siguientes responsabilidades:

- Velar que todos los participantes del proyecto sigan las reglas y proceso de Scrum, encajándolas en la cultura de la organización, y guiar la colaboración del equipo con el cliente de manera que las sinergias sean máximas. Esto implica:
- Asegurar que la lista de requisitos priorizada esté preparada antes de la siguiente iteración.
- Facilitar las reuniones de Scrum (planificación de la iteración, reuniones diarias de sincronización del equipo, demostración, retrospectiva), de manera que sean productivas y consigan sus objetivos. (Palacio, 2014)

Equipo (Team)

Grupo de personas que de manera conjunta desarrollan el producto del proyecto.

Comparten la responsabilidad del trabajo que realizan (así como de su calidad) en cada iteración y en el proyecto. El tamaño del equipo está entre 5 y 9 personas. Por debajo de 5 personas, cualquier imprevisto o interrupción sobre un miembro del equipo compromete seriamente el compromiso que han adquirido y, por tanto, el resultado que se va a entregar al cliente al finalizar la iteración. Por encima de 9 personas, la comunicación y colaboración entre todos los miembros se hace más difícil y se forman subgrupos.

Es un equipo auto gestionado, que realiza de manera conjunta las siguientes actividades:

- Seleccionar los requisitos que se compromete a completar en una iteración, de forma que estén preparados para ser entregados al cliente.
- En la lista de requisitos priorizados del producto, estimar la complejidad de cada uno de ellos.
- En la reunión de planificación de la iteración decide cómo va a realizar su trabajo:
    - Seleccionar los requisitos que pueden completar en cada iteración, realizando al cliente las preguntas necesarias.
    - Identificar todas las tareas necesarias para completar cada requisito.
    - Estimar el esfuerzo necesario para realizar cada tarea.
    - Cada miembro del equipo se asigna a las tareas.
- Durante la iteración, trabajar de manera conjunta para conseguir los objetivos de la iteración. Cada especialista lidera el trabajo en su área y el resto colaboran si es necesario para poder completar un requisito.

Los miembros del equipo deben dedicarse al proyecto a tiempo completo para evitar dañar su productividad por cambios de tareas en diferentes proyectos, para evitar interrupciones externas y así poder mantener el compromiso que adquieren en cada iteración.

15.3. Componentes de la metodología scrum

Definición del proyecto (Product Backlog): Consiste en un documento que recoge el conjunto de requerimientos que se asocian al proyecto. Es responsabilidad del Product Owner realizar esta definición y establecer las prioridades de cada

requerimiento. Es un documento de alto nivel, que contiene descripciones genéricas (no detalladas), y que está sujeto a modificaciones a lo largo del desarrollo.

Definición del Sprint (Sprint Backlog): Un sprint debe entenderse como un subconjunto de requerimientos, extraídas del product backlog, para ser ejecutadas durante un periodo entre 1 y 4 semanas de trabajo. El sprint backlog sería el documento que describa las tareas que son necesarios realizar para abordar los dichos subconjuntos de requerimientos.

Ejecución del Sprint: Sería el periodo de entre 1 y 4 semanas (periodo definido previamente en base a las tareas recogidas en el sprint backlog) durante el cual el equipo de trabajo abordaría las tareas de desarrollo correspondientes. Una vez iniciada la ejecución de un sprint definido, este no podrá ser modificado

Entrega: Una vez concluida la ejecución del sprint, se dispondrá de una porción de la aplicación potencialmente definitiva.

Evolución del proyecto (Burn down): Es un documento que refleja el estado del proyecto, indicando el volumen de requerimientos que en ese momento se encuentran pendientes de ser abordados (en el product backlog), los requerimientos que en ese momento se están desarrollando (sprint backlog). (Diaz, 2014)

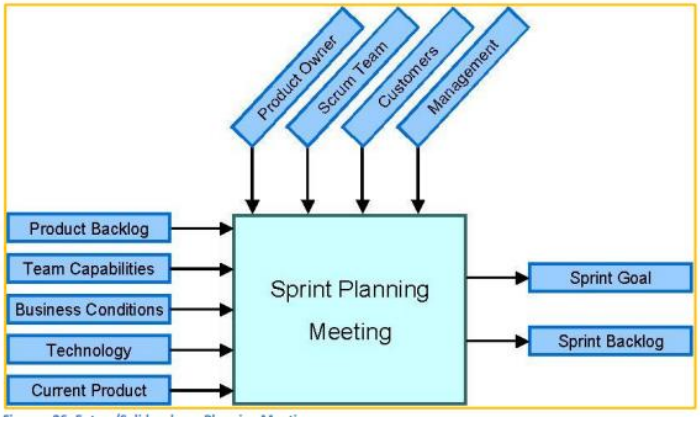

Ilustración 15. Evaluación de sprints

15.4. Reuniones de trabajo en un contexto scrum

Planificación de sprint: Se realiza al principio de cada ciclo de sprint, y está encaminada a seleccionar el conjunto de requerimientos del product backlog que serán abordado, el equipo de trabajo que será necesario y el tiempo que se estima (entre 1 y 4 semanas) para su desarrollo.

Reunión diaria: Conocida como daily scrum, se realiza al comienzo de cada día en que ese esté ejecutando un sprint. Es una reunión corta (no más de 30 minutos) en la que los integrantes del equipo responden las siguientes preguntas:

¿Qué has hecho desde la última reunión?

¿Qué problemas has encontrado para realizar el trabajo previsto?

¿Qué planeas hacer antes de la próxima reunión?

Revisión de sprint: Una vez concluido el ciclo de sprint se mantiene una reunión en la que se define qué parte del trabajo previsto se ha completado y qué parte permanece pendiente. En cuanto al trabajo completado se realiza una revisión (demo) del mismo al product owner y otros usuarios que pudiesen estar involucrados.

Retrospectiva de sprint: Es una reunión en la que todos los miembros del equipo realizan una valoración del trabajo realizado en el último sprint, identificando puntos de mejora de cara a los siguientes a realizar. El objetivo principal es introducir un componente de mejora continua en el proceso. (VASS digital, 2012)

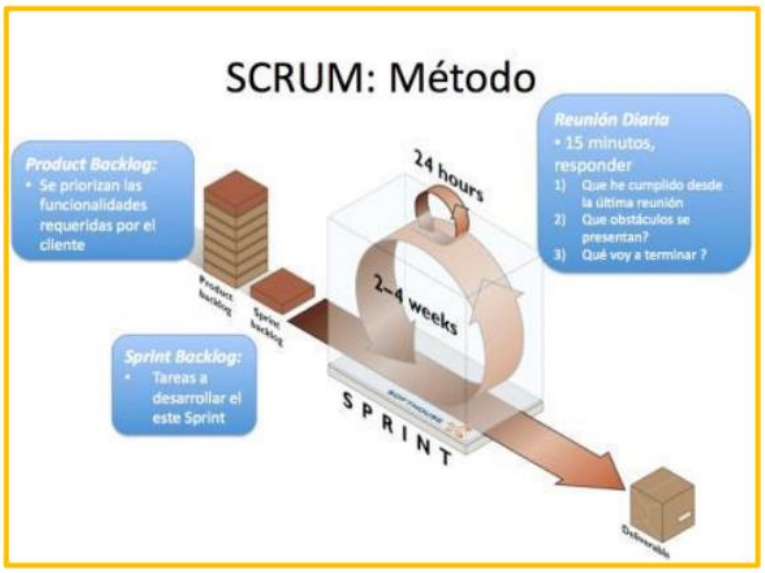

Ilustración 16. Métodos de sprint

SCRUM es una metodología muy dinámica que nos permite tener una mejor relación con el cliente, lo que funciona de manera a que se perfeccionan los detalles de para que al final se entregue un producto de calidad con los menores errores posibles, es precisamente por eso que se piensa como parte del proyecto y así adentrarse en cada una de sus fases para su implementación.

16. Plan de trabajo

16.1. Cronograma de actividades

No.	Actividad	Producto (Evidencia de actividad realizada)	P / R	SEMANAS													
				1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Aperturar estadía - Presentación formal	Hoja de apertura	P														
			R														
2	Hacer levantamiento de requerimientos - Especificar la planeación del proyecto, objetivos, alcance, justificación, etc.	Documento de Word (Tesina)	P														
			R														
3	Comparar el funcionamiento entre aplicaciones para auditoria y escáner de aplicaciones web	Documento de Word	P														
			R														
4	Analizar los productos entregables que se llevan a cabo en el desarrollo de proyectos y adherirlos a la metodología de acuerdo en la fase que corresponda	Presentación de PowerPoint	P														
			R														
5	Elegir una aplicación para la auditoria de seguridad y verificar como funciona cada uno de los módulos disponibles	Documento de Word	P														
			R														
6	Hacer pruebas básicas con aplicaciones exclusivas para	Documento de Word	P														

	pruebas y evaluar el comportamiento de la herramienta		R															
7	Evaluar el documento de procesos que se utiliza hasta el momento y comenzar con las mejoras en cuanto a las fases de la metodología.	Documento de Word	P															
			R															
8	Implementar las mejoras al documento de procesos agregando los puntos importantes que marca la metodología de la cual se hace uso.	Documento de Word	P															
			R															
9	Actualizar los diagramas de comportamiento que se generan para dar explicación grafica de como fluye los procesos y los productos entregables.	Imágenes png	P															
			R															
10	Actualizar los formatos de plantillas para cada producto entregable o crearlo si aún no se ha implementado.	Documentos de Excel	P															
			R															
11	Hacer pruebas reales con las aplicaciones web que han sido desarrolladas en Software Factory y documentarlas.	Documento de Word	P															
			R															
12	Actualizar el repositorio, ordenando cada fase con sus respectivos productos entregables.	Documento de Word (Tesina)	P															
			R															
13	Crear un manual de usuario para indicar el funcionamiento de la herramienta a utilizar, partiendo desde su instalación.	Documentos PDF	P															
			R															

14	Brindar capacitación al equipo de desarrollo para el uso correcto de la herramienta y acerca de los nuevos procesos en la metodología.	Minuta	P															
			R															
15	Cierre de proyecto - Entrega de formato de liberación de estadía.	Hoja de cierre	P															
			R															

16.2. Diseño de estrategias

Para la selección de cada actividad antes planteada en el cronograma de actividades, se organizaron ciertas ideas y estrategias que ayudaron a la determinación de cada actividad, a continuación se presentan las razones por las que se indicó dicha actividad.

1. Apertura de estadía (Presentación formal)

El primer requisito para comenzar con el proyecto es la aceptación de los objetivos y la descripción del mismo, la cual debe ser revisada y validada por un comité especifico y así poder aprobar el proyecto para su posterior desarrollo. Es por ello que se debe hacer entrega de un documento formal que apertura el proyecto como requisito fundamental y oficial tanto para el alumno como para la empresa beneficiada.

Se planteó para una semana debido a que se deben conocer todos los aspectos básicos del proyecto, así también las áreas involucradas con las que se va a interactuar, la estructura organizacional y la base del problema que se pretende resolver.

2. Especificar la planeación del proyecto, objetivos, alcance, justificación, etc.

Esta estrategia forma parte de alguno de los apartados dentro de la tesina que se entrega al final del desarrollo del proyecto, los puntos que se mencionan son de suma importancia para conocer a fondo el proyecto, el ¿por qué?, ¿para qué? y ¿hasta dónde se debe llegar? son las respuestas que proporciona esta parte del documento.

Se propuso realizar la actividad en una semana para que se pueda comprender en un todo la problemática y se brinde un plan de trabajo, el cual debe ser guía primordial a lo largo de su desarrollo. Es importante que este apartado se realice de forma acertada porque es una de las bases más importantes que definirán la viabilidad del proyecto.

3. Comparar el funcionamiento entre aplicaciones para auditoria y escáner de aplicaciones web

Como se sabe, parte del proyecto contempla la implementación de una herramienta para la auditoria de vulnerabilidad de aplicaciones web, pero antes de llevar a cabo la ejecución de dicha herramienta se debe hacer una evaluación entre todas las que el mercado ofrece y al final hacer la elección de la mejor opción.

Se ha indicado está actividad para la tercera semana y posterior comenzar con las actividades de desarrollo, donde se hará una investigación para identificar las características que ofrecen las diversas aplicaciones de auditoria de seguridad de aplicaciones web que hasta el momento se tienen en el mercado. Al término de la actividad ya se deberá trabajar con la funcionalidad de una aplicación específica. Es importante tomarse el tiempo necesario para hacer la elección, dado a que es fundamental para el complemento del trabajo que ya se realiza en Software Factory, el cual debe ser de calidad.

4. Analizar los productos entregables que se llevan a cabo en el desarrollo de proyectos y adherirlos a la metodología de acuerdo a la fase que corresponde.

La actividad tiene relación con la mejora de los procesos de la metodología que siguen en el área de Software Factory, que es parte importante del proyecto a realizar.

En este apartado dentro del cronograma de actividades se pretende analizar cómo funcionan los procesos dentro del área, cómo fluye la información y que productos entregables se crean y cuales son reflejados dentro de la documentación. También es importante hacer una comparación del Cheklist de AMS no SAP (Documento que proporciona dicha área donde enlista los productos entregables que deben formar parte para el traslado del proyecto).

La actividad se indicó en la semana 4 dado a que se tendrán reuniones diarias en dicha semana para conocer un poco de los documentos que se operan y que áreas/persona(s) son responsables de crearlo y así poder ubicarlos en la fase correcta.

5. Elegir una aplicación para la auditoria de seguridad y verificar como funciona cada uno de los módulos disponibles.

La presente actividad en relación con la anterior se pensó porque se deben aclarar algunos puntos en cuanto a las fases de la metodología con todo el equipo en conjunto y requiere realizar una junta en la semana 5 para poder dar fin a la actividad anterior. Es por eso que se dará seguimiento a la implementación de la herramienta de auditoria de seguridad que ya se ha elegido en la semana 3, haciendo interacción con ella y familiarizándose con los módulos que ofrece analizando su función.

6. Hacer pruebas básicas con aplicaciones exclusivas para pruebas y evaluar el comportamiento de la herramienta.

Se buscará hacer pruebas con la herramienta para probar si realmente funcionan todas las características que ofrece, para ello, se hará uso de aplicaciones con cierto número de vulnerabilidades para así poder evaluar si la herramienta es competitiva y da resultados precisos.

7. Evaluar el documento de procesos que se utiliza hasta el momento y comenzar con las mejoras en cuanto a las fases de la metodología.

Hasta el momento se cuenta con un documento guía de procesos que realizan en Software Factory. Este documento debe ser modificado de acuerdo al trabajo realizado anteriormente, agregando los procesos que no se menciona y si se llevan a cabo y viceversa.

Se planteó la actividad para la semana 7 dado a que en esta semana ya se conoce más acerca de los procesos y las áreas que comprende el trabajo que realiza Software Factory y así será más fácil brindar la aportación.

8. Implementar las mejoras al documento de procesos agregando los puntos importantes que marca la metodología de la cual se hace uso.

El implementar las mejoras hace referencia a completar el documento a como es marcado en la metodología SCRUM de la cual hacen uso, tratando de que el documento cubra en un cien por ciento todas las actividades que conforman los proyectos de SWF, para que pueda ser una guía fiable. Las semanas 8, 9 y 10 se tomarán para completar el documento con todas las partes que lo componen.

9. Actualizar los diagramas de comportamiento que se generan para dar explicación grafica de como fluye los procesos y los productos entregables.

Dentro del documento de procesos, se muestran algunos diagramas que reflejan el flujo de las fases o procedimientos y en qué momento se genera un entregable pero, hasta el momento no se hace referencia de todos los productos y flujos, lo que hace que se deban actualizar y crear los que no se muestran al momento.

10. Actualizar los formatos de plantillas para cada producto entregable o crearlo si aún no se ha implementado.

Las plantillas son los productos que se mencionan en la actividad anterior. Aquí se hará la modificación de algunos formatos, optimizándolos de la mejor manera posible y en el caso de que aún no se realicen, se debe crear una plantilla e indicar un nombre adecuado del formato, así mismo agregarlo al repositorio.

11. Hacer pruebas reales con las aplicaciones web que han sido desarrolladas en Software Factory y documentarlas.

La actividad número 6 puede parecerse a la presente, pero está hace referencia a pruebas dadas a las aplicaciones que desarrolla meramente el área SWF, con las cuales se establecerá una métrica para definir el límite de vulnerabilidad posible en las aplicaciones y como se va a evaluar si se están desarrollando aplicaciones de calidad.

12. Actualizar el repositorio, ordenando cada fase con sus respectivos productos entregables.

Después de ya haber organizado todos los productos que se generan en cada fase y documentarlos, procede actualizar el repositorio del que hace uso el equipo de trabajo, con ello el equipo podrá identificar como se debe trabajar y que productos abarca cada fase.

13. Crear un manual de usuario para indicar el funcionamiento de la herramienta a utilizar, partiendo desde su instalación.

Para finalizar con la actividad del funcionamiento de la herramienta de auditoria de seguridad de aplicaciones web, se debe crear un manual de instalación y un manual de usuario para que todo el equipo de desarrollo pueda tomarlo como apoyo para su uso. Se planteó esta actividad para la semana 13 debido a que ya se han finalizado las pruebas y se ha establecido una métrica.

14. Brindar capacitación al equipo de desarrollo para el uso correcto de la herramienta y acerca de los nuevos procesos en la metodología.

Se convocará a mínimo 3 juntas para capacitar al equipo de desarrollo y resolver dudas que se presenten.

15. Cierre de proyecto - Entrega de formato de liberación de estadía.

Se libera el formato de cierre de estadía, evaluando los entregables del cronograma de actividades.

CAPÍTULO IV: ACTUALIZACIÓN DE METODOLOGÍA DE SW

17. Implementación de la metodología Scrum

Software Factory actualmente implementa la metodología SCRUM para el desarrollo de sus proyectos, se basa en las buenas practicas que está metodología ofrece y se rige por las funciones que son óptimas para cumplir adecuadamente con los procesos.

Se llevan a cabo los puntos más resaltados dentro de la metodología, es decir, se hacen reuniones diarias para resolver dudas o compartir puntos de vista, se trabaja mediante sprints que se revisan cada 4 semanas, se hacen reuniones de scrum póker para elegir la prioridad de cada requerimiento, se revisa cada sprint con los usuarios finales y el product owner, etc.

Las fases del ciclo de vida del proyecto que se implementan son las siguientes:

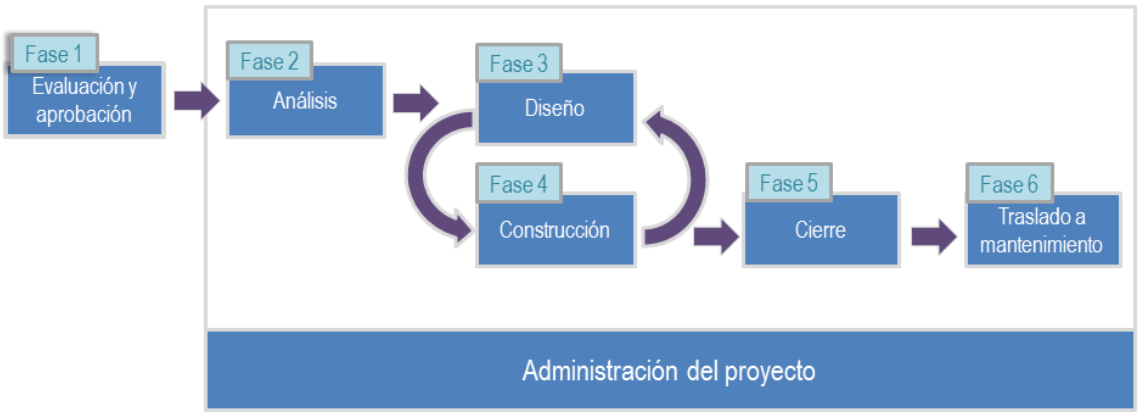

Ilustración 17. Fases del ciclo de vida de los proyectos

Evaluación y aprobación: Esta es la fase en la que se decide si se realiza el proyecto o no. El área interesada entrega un documento el cual denominan “Lastenheft” donde se describen los requerimientos que se desean contenga el nuevo sistema, además de justificar el porqué de la necesidad de realizarlo. Cuando el proyecto es aceptado por el

área de Software Factory y el área de negocio, este pasa a un estado de administración del proyecto, como se puede ver en la imagen, donde la siguiente fase es el análisis.

Análisis: Aquí se plantea como se llevará a cabo el proyecto, el tiempo que se dispondrá para realizarlo y lo más relevante, que son las historias de usuario que se describen en esta fase para tener un panorama más claro de la dificultad y grandeza del proyecto.

Diseño: También conocido como Sprint 0, en esta fase se diseñan algunos componentes de la aplicación establecidos en una aplicación base que funciona como la guía de cada proyecto. Esta fase interactúa directamente con la siguiente que es la de construcción, dado a que se pueden hacer cambios o trabajar en las dos al mismo tiempo.

Construcción: Fase en la que se llevan a cabo los sprints, como su nombre lo dice es la construcción de la aplicación donde se realiza la programación de las historias de usuario. Se requieren reuniones cada cuatro meses con el product owner y el usuario final para ver y aprobar el rendimiento del equipo. Por cada proyecto se hace un estimado de 5 sprints. Las pruebas funcionales de las aplicaciones también se llevan a cabo en esta fase.

Cierre: Aquí los usuarios repasan la aplicación y si cumple con los requerimientos y el objetivo establecido la aprueba. También se hace la liberación a productivo donde se evalúa el comportamiento de la aplicación. La aplicación ya debe estar lista para esta fase.

Traslado a mantenimiento: Es en esta fase donde se ocasionan algunos problemas que atrasan la entrega de la aplicación, debido a que para trasladarla a mantenimiento se deben hacer una cierta cantidad de entregas, de capacitaciones y acuerdos, para que al final la responsabilidad del funcionamiento de la aplicación pase a manos de AMS no SAP.

18. Estructuración de las fases de la metodología

En Fábrica de Software actualmente se implementa la metodología SCRUM con 6 fases durante el desarrollo del proyecto. Para comenzar con la nueva estructuración, se debe identificar en que momento es factible entregar cada producto, para que al final no se generen la mayoría de ellos en la última fase, lo cual atrasa más la liberación de la aplicación.

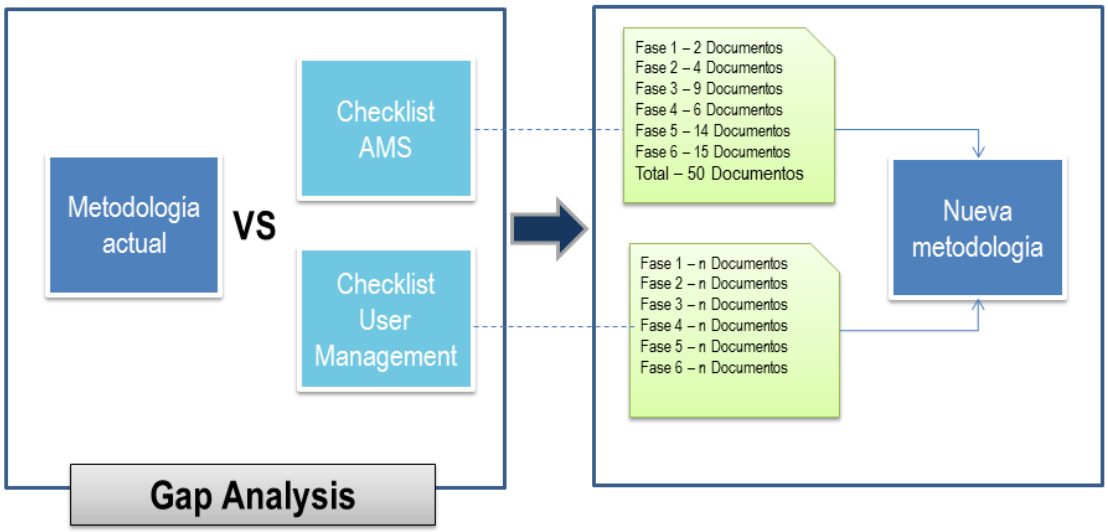

Ilustración 18. Comparativa de metodología actua y nueva metodología

La imagen que se muestra en la parte superior refleja el estado de la metodología actual y la comparación con los checklists de AMS no SAP e User Management, debido a que al final del desarrollo de la aplicación, esta pasa a mantenimiento, donde los encargados de dar soporte y mantenimiento son dichas áreas. Ellos entregan un checklist de todos los documentos, correos y evidencias que requieren para aprobar que el desarrollo del proyecto haya concluido, además de que se debe hacer entrega del código fuente, donde Software Factory termina su responsabilidad con el proyecto

CAPÍTULO V: HERRAMIENTA DE AUDITORIA DE SEGURIDAD

19. Comparativa de herramientas de auditoria de seguridad

La seguridad en la web como bien sabemos es muy vulnerable ante personas malintencionadas que solo buscan dañar o robar la información contenida dentro de la web. Muchos desarrolladores tratan de cubrir todas estas posibles brechas de vulnerabilidad dentro de sus sitios web, pero en ocasiones es muy complicado identificar todos los posibles ataques que se pudieran presentar, es aquí donde los "Web Application Security Scanner" toman el papel para ayudar a encontrar todas las posibles vulnerabilidades de seguridad realizando las pruebas de manera automática en una aplicación web. Los escáneres no tienen acceso al código fuente, sólo se realizan pruebas funcionales y tratan de encontrar vulnerabilidades de seguridad. A continuación se hace una comparativa de algunos escáneres de los cuales se debe hacer una elección de acuerdo a los puntos resaltados en la tabla.

COMPARATIVA DE ESCÁNERES DE SEGURIDAD DE APLICACIONES WEB										
Nombre	**Portátil**	**Interfaz Grafica**	**Rápida**	**Recomendada para:**		**Usabilidad**	**Informe PDF**	**Desarrollada en:**	**Open-Source**	**Gratuita**
				Grandes App's	**Pequeñas App's**					
Grabber	☑	☐	☐	☑	☐	☐	☐	Python	☑	☑
Wapiti	☑	☐	☑	☑	☑	☑	☑	Python	☐	☑
Ataque Zed Proxy	☑	☑	☑	☑	☑	☑	☐	Java	☑	☑
Arachni	☑	☑	☑	☑	☑	☑	☐	Ruby	☑	☑
OpenVAS	☐	☑	☑	☑	☑	☐	☐	Java	☑	☑

Ilustración 19. Tabla comparativa de escáneres de seguridad

Grabber

Grabber es un escáner de aplicaciones web agradable que puede detectar muchas vulnerabilidades de seguridad en aplicaciones web. Se realiza exploraciones y muestra en donde está el error. Esta herramienta es capaz de detectar las siguientes vulnerabilidades:

- Cross Site Scripting
- Inyección SQL
- Pruebas de Ajax
- La inclusión de archivos
- JS analizador de código fuente
- Comprobación del archivo de copia de seguridad

No es rápido en comparación con otros escáneres de seguridad, pero es simple y portátil. Esto se debe utilizar sólo para probar pequeñas aplicaciones web, ya que toma demasiado tiempo para escanear grandes aplicaciones.

Esta herramienta no ofrece ninguna interfaz gráfica de usuario. Tampoco puede crear cualquier informe en PDF. Esta herramienta fue diseñada para ser simple y para uso personal. Se puede probar esta herramienta sólo para uso personal. Si se está pensando en él para el uso profesional, no se recomienda.

Esta herramienta fue desarrollada en Python y también tiene una versión ejecutable (.exe) . El código fuente está disponible, por lo que se puede modificar de acuerdo a las necesidades individuales. El script principal es grabber.py, que una vez ejecutado llamadas otros módulos como sql.py, xss.py u otros.

Wapiti

Wapiti es un escáner de vulnerabilidades de aplicaciones web, que le permite auditar la seguridad de sus aplicaciones web. Realiza exploraciones de "caja negra", es decir,

no estudia el código fuente de la aplicación, sino que analiza las páginas web de la aplicación web desplegada, buscando secuencias de comandos y formularios donde pueda inyectar datos.

Una vez que recibe esta lista, Wapiti actúa como un fuzzer, inyectando cargas útiles para ver si un script es vulnerable.

Primero escribimos sobre Wapiti WAYYY en 2006 - Wapiti - Aplicación web de escáner / Black-box de prueba. Ha recorrido un largo camino desde entonces, pero no parece haber mucho desarrollo activo desde 2013 - que es una pena, ya que es una buena herramienta.

Wapiti puede detectar las siguientes vulnerabilidades:

- Divulgación de archivos (local y remoto include / require, fopen, readfile ...)
- Inyección de base de datos (Inyecciones PHP / JSP / ASP SQL e Inyecciones XPath)
- Inyección XSS (Cross Site Scripting) (reflejada y permanente)
- Comando Detección de ejecución (eval (), system (), passtru () ...)
- Inyección de CRLF (división de respuesta HTTP, fijación de sesión ...)
- Inyección XXE (XmleXternal Entity)
- Uso de archivos potencialmente peligrosos (gracias a la base de datos Nikto)
- Configuraciones .htaccess débiles que pueden ser omitidas
- Presencia de archivos de copia de seguridad que proporcionan información confidencial (divulgación del código fuente)
- Esta excelente herramienta programada en Python destaca por sus detallados informes, donde se puede ver la descripción del problema encontrado, cómo resolver dicho problema y algunas páginas de referencia donde podemos ampliar nuestro conocimiento sobre el tema.

Wapiti es un escáner de vulnerabilidades para aplicaciones web, licenciado bajo la GPL v2 , que busca fallos XSS, inyecciones SQL y XPath, inclusiones de archivos (local y remota), ejecución de comandos, inyecciones LDAP, inyecciones CRLF, para que se ponga a prueba la seguridad de nuestras aplicaciones web y podamos corregirlas.

Las características con las que cuenta Wapiti son las siguientes:

- Genera reportes de vulnerabilidad en varios formatos (HTML, XML, JSON, TXT…)
- Puede suspender y reanudar una exploración o un ataque
- Puede darle colores en el terminal para resaltar las vulnerabilidades
- Diferentes niveles de verbosidad
- Manera rápida y fácil de activar / desactivar los módulos de ataque
- Agregar una carga útil puede ser tan fácil como añadir una línea a un archivo de texto
- Soporta métodos GET y POST HTTP para ataques
- También soporta multipart y puede inyectar cargas útiles en nombres de archivos (subir)
- Puede mostrar una advertencia cuando se detecta una anomalía (por ejemplo, 500 errores y tiempos muertos)
- Hace la diferencia entre vulnerabilidades XSS permanentes y reflejadas.

Es importante mencionar que el uso de este tipo de herramientas, que automatizan el testeo de vulnerabilidades en tus aplicativos web, ayudan a encontrar fácilmente algunos problemas de las aplicaciones web, pero no reemplazan una auditoria concienzuda del código fuente.

Ataque Zed Proxy (ZAP)

ZAP es una poderosa herramienta para realizar ataques de penetración (disciplina conocida como Pentesting), que permite analizar sitios web para buscar sus vulnerabilidades, con muy diversos fines, como por ejemplo, aprendizaje o securización del sitio. Inicialmente basado en el código de Paros Proxy (una de las herramientas de pentesting más usadas en su momento y actualmente abandonada), ZAP se basa en la lista OWASP de las vulnerabilidades web más comunes para su desarrollo, por lo que incluye una gran cantidad de herramientas capaces de detectar casi cualquier vulnerabilidad que pueda existir en un sitio web. Además, ZAP es una herramienta gratuita, multiplataforma, muy orientada a la comunidad (cualquiera puede aportar su granito de arena al proyecto, desde solicitando nuevas herramientas hasta incluso participar en su desarrollo), y cuya intención es hacer accesible el pentesting a todo el mundo.

Las principales características de ZAP son:

- Herramienta totalmente gratuita y de código abierto.
- Herramienta multi-plataforma, compatible incluso con Raspberry Pi.
- Fácil de instalar, dependiendo únicamente de Java 1.7 o superior.
- Posibilidad de asignar un sistema de prioridades.
- Traducida a más de 12 idiomas, entre ellos, el español.
- Excelente manual de ayuda y gran comunidad en la red.

Los usuarios de esta herramienta forense de seguridad podrán auditar diferentes aplicaciones web con una serie de funciones y análisis específicos:

- Posibilidad de comprobar todas las peticiones y respuestas entre cliente y servidor.
- Posibilidad de localizar recursos en un servidor.

- Análisis automáticos.
- Análisis pasivos.
- Posibilidad de lanzar varios ataques a la vez.
- Capacidad para utilizar certificados SSL dinámicos.
- Soporte para utilizar tarjetas inteligentes (DNI-e, por ejemplo) y certificados personales.
- Análisis de sistemas de autenticación.
- Posibilidad de actualizar la herramienta automáticamente.
- Dispone de una tienda de extensiones (plugins) con las que añadir más funcionalidades a la herramienta.

Autenticación y gestión de sesiones.

Proporciona un API Rest (disponible en JSON, HTML y XML) para la integración con otras herramientas, que permite el uso de las funcionalidades de escaneo activo y spider, aunque en futuras versiones de ZAP se aumentara el número de funcionalidades disponibles a partir del API.

Actualizaciones automáticas.

Plugins integrados y un Marketplace de plugins en constante actualización.

Se ha escogido la herramienta OWASP Zed Attack Proxy por ser una de las pocas herramientas que facilitan su integración con otras aplicaciones al disponer sus funcionalidades en un API Rest. De esta manera, la aplicación puede delegar funcionalidades sin acoplar su implementación a una herramienta en concreto.

Además es una herramienta en constante actualización desarrollada por un organismo importante dentro del campo de la seguridad informática.

Arachni

Arachni es un framework modular y de alto rendimiento está desarrollado en Ruby y sirve para escanear vulnerabilidades en un sitio web. Está pensado para ayudar a los profesionales en los análisis y pruebas de penetración, también es muy útil para administradores de servidores o webmasters que evalúan la seguridad de las aplicaciones web modernas.

Arachni es de código libre y gratuito, es multiplataforma, compatible con todos los principales sistemas operativos Windows, Mac OS y Linux.

Este framework tiene una interface web pero sus librerías son módulos independientes por lo que se pueden correr en forma separada al mismo tiempo. Es muy bueno ya que su diccionario de vulnerabilidades conocidas puede cubrir una gran cantidad vulnerabilidades. Las herramientas que nos brinda van desde una simple utilidad de escáner de línea de comandos, hasta escáneres de alto rendimiento como inyecciones SQL.

El entorno de navegador integrado mediante su interface web, permite soportar aplicaciones web altamente sofisticadas que utilizan tecnologías como JavaScript, HTML5, la manipulación del DOM y AJAX.

Los administradores pueden configurar la herramienta para limitar los controles pasivos de los archivos a los números de tarjetas de crédito, números de Seguro Social, direcciones IP privadas y otros tipos de información. Desde su máquina cliente, pueden escanear un solo servidor o varios servidores que se ejecutan en Windows, Solaris y/u otros sistemas operativos. Los administradores que actúan como examinadores de penetración y que utilizan la herramienta como parte de su conjunto de herramientas de evaluación están exentos de la exigencia de una licencia para el uso comercial.

Una característica realmente interesante se corresponde con la utilización de diferentes perfiles con los que cuenta Arachni. Entre ellos, se incluyen el perfil por defecto que apunta a un análisis general. Además existe un perfil que, específicamente, realiza análisis sobre vulnerabilidades de tipo XSS (Cross Site Scripting) y uno para SQL Injection. Además, Arachni permite generar un nuevo perfil de acuerdo a las necesidades del auditor.

OpenVas

Los administradores que trabajaron con Nessus deben encontrarlo similar a OpenVAS (Sistema de Evaluación de Vulnerabilidades Abiertas). OpenVAS fue creado como una herramienta gratuita a partir de la última versión gratuita de Nessus en 2005. Ambos utilizan varios escáneres para descubrir vulnerabilidades en servidores desde una máquina cliente. Pero esto no significa que Nessus y OpenVAS pueden apuntar a los mismos tipos de vulnerabilidad.

Para instalar OpenVAS-8, el administrador puede elegir un paquete de terceros o compilar el código fuente. El escáner realiza actualizaciones diarias para las pruebas de vulnerabilidad de red (NVTs) a través del RSS OpenVAS NVT o a través de un servicio de fuentes comerciales. El administrador trabaja con el Asistente de Seguridad Greenbone, un servicio web delgado con una interfaz de usuario para navegadores web.

Coordinando con el módulo de servicio OpenVAS Manager, el administrador gestiona las cuentas de usuario, y puede cambiar las reglas para reiniciar el número de escáneres OpenVAS que pueden apuntar a las redes y/o hosts específicos de sistemas operativos al mismo tiempo. El administrador también puede cambiar las configuraciones para permitir que un usuario escanee su propio host o sincronice manualmente su propio repositorio NVT con una fuente OpenVAS NVT. Él o ella

pueden controlar los horarios de escaneo y mantener una base de datos SQL, como MySQL o SQLite, de configuraciones de escaneo y resultados.

De acuerdo al cuadro comparativo y a la descripción de las características de cada una de las herramientas para la auditoria de seguridad para aplicaciones web, se ha llegado a la conclusión de que Arachni cumple con los requisitos que SWF necesita para el desarrollo de sus aplicaciones web.

20. Funciones de Arachni

Muchas empresas trabajan con aplicaciones web de forma muy estrecha, donde numerosas veces forma parte del modelo de negocio. Debido a esto, las aplicaciones web suelen ser un nodo crítico dentro del esquema tecnológico de la compañía y deben contemplarse los riesgos existentes sobre las mismas. Es por eso que se hace necesario realizar los controles necesarios para estar preparados contra posibles ataques. Arachni es una herramienta muy completa que permite realizar diversos controles y pruebas sobre una aplicación web en particular y obtener información en un cómodo reporte.

Es una herramienta que permite realizar auditorías de seguridad sobre aplicaciones web con un framework Ruby modular y de alto rendimiento, destinado a ayudar a los probadores de penetración y a los administradores a evaluar la seguridad de las aplicaciones web modernas. Es gratis, con su código fuente público y disponible para su revisión.

Es multiplataforma, soporta todos los principales sistemas operativos (MS Windows, Mac OS X y Linux) y se distribuye a través de paquetes portátiles que permiten un despliegue instantáneo.

Es lo suficientemente versátil como para cubrir una gran cantidad de casos de uso, desde una sencilla utilidad de escáner de línea de comandos, hasta una red global de alto rendimiento de escáneres, una biblioteca Ruby que permite realizar auditorías de scripts, una plataforma de colaboración web multi-usuario multi-scan. Además, su sencilla API REST hace que la integración sea fácil. También es importante mencionar que debido a su entorno de navegador integrado, puede soportar aplicaciones web altamente complicadas que hacen uso intensivo de tecnologías como JavaScript, HTML5, manipulación de DOM y AJAX.

Arachni ofrece una interfaz gráfica para hacer más fácil su manejo e implementación, mediante la interfaz, podemos encontrar un menú con diversas opciones que darán más herramientas para llevar a cabo el análisis de manera factible.

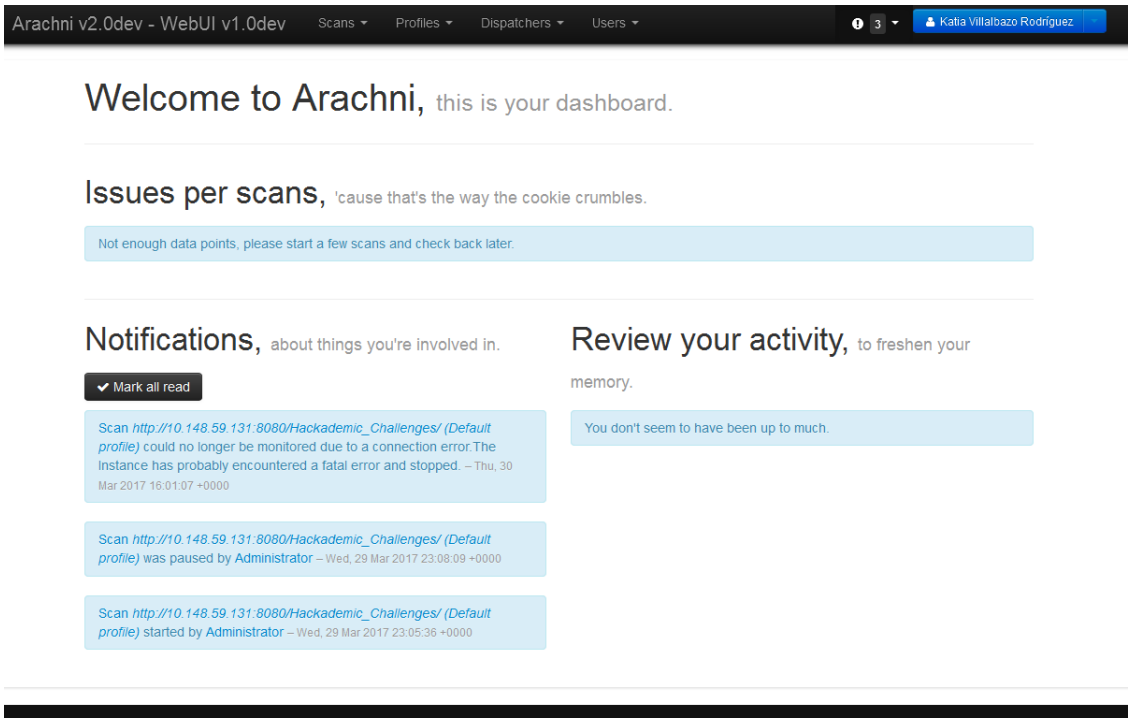

Ilustración 20. Interfaz de Arachni

Arachni no solo nos permite realizar un análisis de vulnerabilidad, sino también poder configurarlo de acuerdo a las características que sean más convenientes en cuanto a la aplicación que se va a auditar. Esto puede depender de la plataforma de desarrollo,

desde la base de datos, el servidor web, el lenguaje de programación y/o los frameworks que se han implementado, dado a que algunas vulnerabilidades solo aplican para ciertos entornos de desarrollo y estaría de más evaluarlas en estos casos; permite elegir solo la evaluación de vulnerabilidades específicas de acuerdo al criterio del evaluador. También en la configuración de perfil se pueden agregar plugins que ofrece la herramienta, para añadir funcionalidad adicional al sistema de forma modular, de esta manera el núcleo sigue siendo el mismo y hace que sea fácil para cualquier persona agregar funcionalidades propias o de terceros.

20.1. Vulnerabilidades Activas

A continuación se describe lo más breve posible las vulnerabilidades que detecta Arachni, dando una explicación de cada una de ellas y como afectan la seguridad. Arachni divide las vulnerabilidades por Activas y Pasivas, las siguientes hacen referencia a la primera.

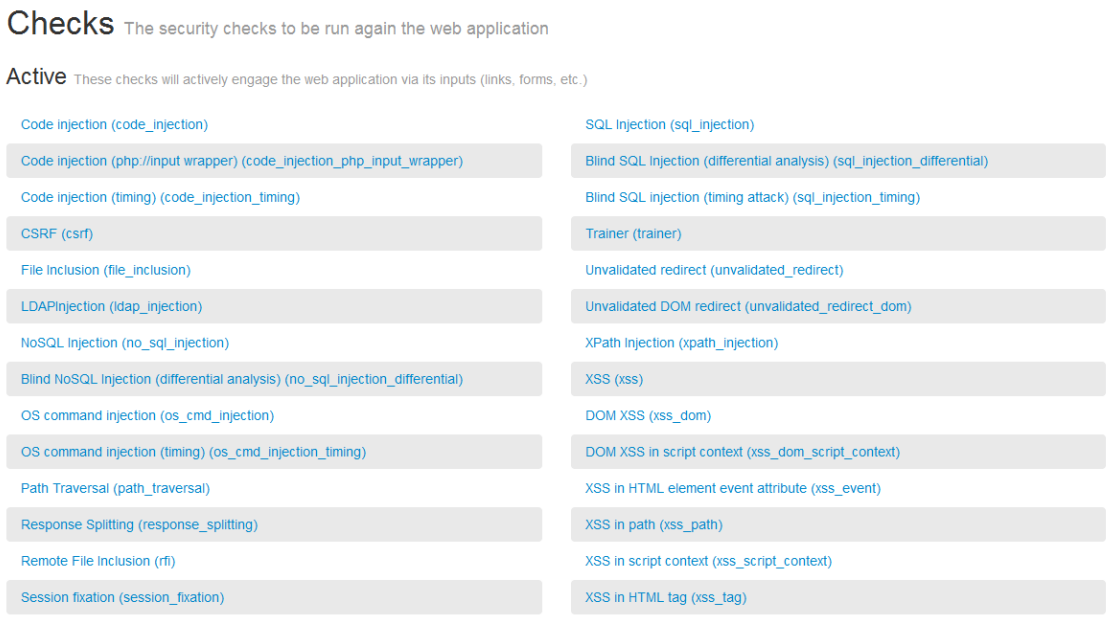

Ilustración 21. Vulnerabilidades Activas

Inyección de código. Es un tipo de ataque que consiste en que los atacantes extraigan información, roben credenciales, tomen control de la página atacada o algún otro tipo de actividades maliciosas.

Este tipo de amenazas incluye las siguientes técnicas conocidas que arachni también incluye entre todas sus opciones:

Inyección SQL. Este ataque se aprovecha de una vulnerabilidad en que una variable o componente de una página que utiliza código SQL (código que accede bases de datos) permite que un atacante emplee comandos para manipular datos y acceder información confidencial.

XSS o Cross Site Scripting. En español, secuencias de comandos en sitios cruzados. Este ataque aprovecha vulnerabilidades en las páginas de Internet que permiten que un cibercriminal inserte código de JavaScript o lenguajes similares para inyectar código HTML en las páginas que atacan, alterando así la funcionalidad de la misma.

CSRF o Cross Site Request Forgery. En español, falsificación de petición en sitios cruzados. Este ataque consiste en que el navegador de Internet de una víctima es forzado a enviar una solicitud a una página de Internet a la que la víctima tiene acceso o a la que ya se le ha autorizado su acceso. Por ejemplo: si tú haces clic en un enlace malicioso, y la página de Internet de tu banco dejó una cookie autorizándote acceso, el enlace malicioso podría enviar una solicitud a la página de tu banco. Puedes prevenir este tipo de ataques si sales (logout) de las páginas bancarias después de haberlas utilizado.

RFI o Remote File Inclusion. En español, inclusión remota de archivos. Este ataque existe en páginas escritas en el lenguaje PHP que están escritas de forma descuidada y que permiten el enlace a archivos ubicados en otras computadoras. La inclusión de archivos pudiera resultar en ejecución de código malicioso.

En los últimos años el ataque más común ha sido el de inyección SQL, que ha sido usando por grupos de hackers famosos, como LulzSec o Anonymous. Otro ataque que ha crecido significativamente en los últimos años es el de XSS.

Path Traversal. Es un ataque por vulnerabilidad que intenta explotar una vulnerabilidad informática que ocurre cuando no existe suficiente seguridad en cuanto a la validación de un usuario, permitiéndole acceder a cualquier directorio superior sin ningún control. La finalidad de este ataque es ordenar a la aplicación a acceder a un archivo al que no debería poder acceder o no debería ser accesible, típicamente el /etc/passwd. Este ataque también es conocido como escalado de directorios.

HTTP Response Splitting. Se produce cuando los datos ingresan a una aplicación web a través de una fuente no confiable, lo más frecuentemente una solicitud HTTP. Los datos se incluyen en un encabezado de respuesta HTTP enviado a un usuario web sin ser validado para caracteres maliciosos.

HTTP Response Splitting. Es un medio para un fin, no un fin en sí mismo. En su raíz, el ataque es directo: un atacante pasa datos maliciosos a una aplicación vulnerable y la aplicación incluye los datos en un encabezado de respuesta HTTP. Para montar un exploit exitoso, la aplicación debe permitir la entrada que contenga caracteres CR (retorno de carro, también dado por% 0d o \ r) y LF (línea de alimentación, también dada por% 0a o \ n) en el encabezado Y la plataforma subyacente debe ser vulnerable a la inyección de tales caracteres. Estos caracteres no sólo le dan a los atacantes el control de los encabezados restantes y el cuerpo de la respuesta que la aplicación pretende enviar, sino que también les permiten crear respuestas adicionales enteramente bajo su control.

Session Fixation. En español, fijación de sesión. Es un método de Session hijacking (robo de sesión) un poco especial, ya que, si normalmente en el robo de sesión se intenta conseguir el identificador de sesión de un usuario ya autenticado, la fijación de

sesión se basa en asignar un identificador de sesión conocido a un usuario antes de que se autentique.

Redireccionamiento invalido. Los redireccionamientos y reenvíos no validados son posibles cuando una aplicación web acepta entrada no fiable que podría hacer que la aplicación web redirija la solicitud a una URL contenida en una entrada no fiable. Al modificar la entrada de URL no fiable a un sitio malicioso, un atacante puede iniciar con éxito una estafa de phishing y robar las credenciales de usuario. Dado que el nombre del servidor en el vínculo modificado es idéntico al sitio original, los intentos de phishing pueden tener una apariencia más confiable. Los ataques de redireccionamiento y de reenvío no validados también se pueden usar para crear una URL maliciosa que pase la comprobación del control de acceso de la aplicación y luego reenvíe al atacante a funciones privilegiadas a las que normalmente no podrían acceder.

Entidad externa de XML. Desde el punto de vista de un atacante, esta característica de XML es muy interesante ya que si una aplicación recibe algún documento XML como entrada, lo procesa; un atacante podrá enviar un documento XML en el cual defina una entidad externa referenciando a cualquier fichero del servidor o recurso de la red interna y volcar el contenido del mismo en la salida. Permitiéndole acceder a cualquier recurso que el servidor de aplicaciones tenga permiso para leer.

20.2. Vulnerabilidades Pasivas

Las vulnerabilidades que se describieron son las más destacadas dentro de la sección Activas que ofrece Arachni, ahora se mostrará una descripción de las vulnerabilidades más importantes dentro de las Pasivas que, prácticamente evalúan las características que podrían necesitar la aplicación web para que sea más segura.

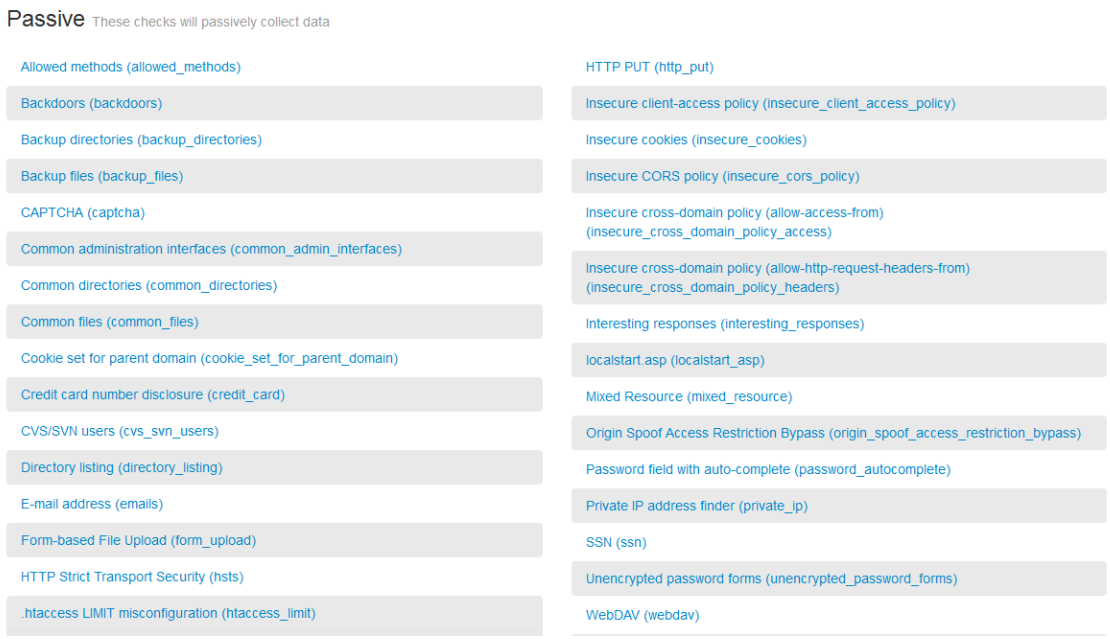

Ilustración 22. Vulnerabilidades Pasivas

Allowed methods. Comprueba los métodos HTTP admitidos.

CATPCHA. Verifica páginas para formularios con CAPTCHAs. CAPTCHA es un acrónimo en inglés para Completely Automated Public Turing test to tell Computers and Humans Apart, que en español se puede traducir como “Prueba de Turing completamente pública y automática para diferenciar máquinas de humanos”.

Típicamente CAPTCHA consiste en que una persona proporcione una serie de caracteres distorsionados que se le muestran en pantalla (similares a los que se muestran en la ilustración), de tal forma que solamente [en teoría] un humano pueda interpretarlos y no una máquina.

Credit card number disclosure. Analiza las páginas de los números de tarjetas de crédito.

.htaccess Limit misconfiguration. Comprueba la configuración incorrecta en las directivas LIMIT que bloquea las solicitudes GET pero permite POST.

HTML Objects. Registra la existencia de etiquetas de objeto HTML. Ya que Arachni no puede ejecutar cosas como Applets de Java y Flash esto sirve como un heads-up al probador de penetración para revisar los objetos en cuestión usando un método diferente.

HTTPOnly cookies. Registra las cookies que son accesibles a través de JavaScript. Una técnica muy útil que existe desde hace tiempo en el navegador Internet Explorer 6 SP1 y adoptada por todas las nuevas versiones de otros navegadores (Firefox desde 3.01, Opera desde 9.5) consiste en declarar los cookies como HTTPonly, protegiéndolos de lectura y escritura por parte de scripts del lado del usuario. De esta forma solo el servidor y el navegador tendrán acceso a la información guardada en los mismos. La solución es en realidad muy simple en algunos casos, por ejemplo en ASP.NET solo tendrá que agregar un parámetro a la línea de configuración "httpCookies" al archivo de configuración web.config.

Interesting response. Los códigos de estado de respuesta HTTP exóticos pueden proporcionar información útil sobre el comportamiento de la aplicación web y ayudar con la prueba de penetración.

Password field with auto-complete. En las aplicaciones web basadas en formularios típicos, es una práctica común para los desarrolladores permitir autocompletar dentro del formulario HTML para mejorar la usabilidad de la página. Con el autocompletar activado (predeterminado), el navegador puede almacenar en caché valores de formulario introducidos previamente.

Para fines legítimos, esto permite al usuario volver a ingresar los mismos datos rápidamente al completar el formulario varias veces. Cuando se habilita el autocompletar en ambos campos de nombre de usuario y contraseña, esto podría permitir a un ciberdelincuente que tenga acceso a la computadora de la víctima, la

posibilidad de que las credenciales de la víctima se introduzcan automáticamente cuando el ciberdelincuente visite la página afectada.

Private IP address finder. Las direcciones IP privadas o no enrutables se utilizan generalmente en una red doméstica o de empresa y normalmente no se conocen por nadie fuera de esa red. Los ciberdelincuentes tratarán de identificar el intervalo de direcciones IP privadas que utiliza su víctima para ayudar a recopilar información adicional que podría dar lugar a un posible compromiso.

20.3. Plugins de Arachni

Arachni también ofrece alguno plugins que hacen más amenas las funciones de la aplicación. A continuación se enlistaran los plugins permitidos que se pueden elegir en la creación de un perfil al igual que las vulnerabilidades.

AutoLogin: Busca el formulario de inicio de sesión en la URL proporcionada por el usuario, combina sus campos de entrada con los parámetros suministrados por el usuario y establece las cookies de la respuesta y la solicitud como cookies de todo el marco.

Si el formulario de inicio de sesión está oculto por defecto y requiere una secuencia de interacciones de DOM para ser visible, este complemento no podrá enviarlo.

AutoThrottle: Supervisa los tiempos de respuesta HTTP y estrangula automáticamente la concurrencia de solicitudes para mantener la estabilidad y evitar matar al servidor.

Beep Notify: Emite un sonido cuando finaliza la exploración. Esto pensado a que las aplicaciones más complejas pueden tardar demasiado tiempo para completar el análisis.

Content-types: Registra los tipos de contenido de las respuestas del servidor. Puede ayudarlo a clasificar e identificar los tipos de archivos disponibles públicamente que a su vez puede ayudarle a identificar archivos que se filtraron accidentalmente.

Cookie Collector: Monitorea y recopila las cookies mientras establece un cronograma de cambios. Muy desalentado cuando la auditoría incluye cookies. Se registrará miles de resultados que conducen a un gran informe, el consumo de memoria muy aumentado y el uso de la CPU.

Discovery-Check response anomalies: Analiza los resultados del análisis e identifica los problemas registrados mediante comprobaciones de descubrimiento (es decir, las comprobaciones que buscan determinados archivos y carpetas en el servidor), mientras que las respuestas del servidor exhibían un factor anómalo de similitud. Hay una buena probabilidad de que estos problemas sean falsos positivos.

E-mail notify: Envía una notificación (y opcionalmente un informe) sobre SMTP al final de la exploración.

Exec: Llama a ejecutables externos en diferentes etapas de escaneo.

Headers Collector: Intercepta las respuestas HTTP y los encabezados de los registros cuyo nombre coincide con los criterios especificados.

- Los nombres de los encabezados serán en minúsculas.
- Si no se han proporcionado patrones, se registrarán todos los encabezados de respuesta.

Health map: Genera una simple lista de direcciones URL seguras / inseguras.

Metrics: Captura métricas sobre varios aspectos de la exploración y la aplicación web.

Proxy: Recopila datos basados en las acciones de los usuarios e intercambia tráfico HTTP y los envía a la cola de páginas del framework para que sean auditados.

Actualiza las cookies de la estructura con las cookies de las solicitudes y respuestas HTTP, por lo que también se puede utilizar para iniciar sesión en una aplicación web.

Soporta intercepción SSL.

Autorización a través de un token de sesión configurable.

Restrict to DOM state: Restringe la exploración al estado DOM de una sola página.

Timing attack anomalies: Analiza los resultados del análisis y registra los problemas que utilizaron los ataques de sincronización mientras las páginas web afectadas mostraron un tiempo de respuesta inusualmente alto; Una situación que hace que los problemas registrados no sean concluyentes o (posiblemente) falsos positivos.

Las páginas con tiempos de respuesta altos suelen incluir procesamiento de alta capacidad, lo que los convierte en objetivos principales de los ataques de denegación de servicio.

Uncommon headers: Intercepta respuestas HTTP y registra encabezados infrecuentes.

Uniformity: Analiza los resultados del análisis y registra los problemas que persisten en diferentes páginas. Esto suele ser una señal de falta de un punto central / único de desinfección de entrada, una mala práctica de codificación.

Vector Collector: Analiza cada página y recopila información sobre los vectores de entrada. Registrará miles de resultados que conducirán a un informe enorme, un consumo de memoria mucho mayor y un uso de la CPU.

WAF Detector: Realiza perfiles básicos en la aplicación web para evaluar la existencia de un firewall de aplicaciones web.

Este es un proceso de 4 etapas:

1. Agarra la página original tal como está.
2. Envíe un montón de cuerdas inocentes (vainilla) en entradas inexistentes para perfilar el comportamiento normal.
3. Envíe un montón de secuencias sospechosas (picante) en entradas inexistentes y compruebe si el comportamiento cambia.
4. Hacer cabezas o colas de las respuestas recogidas.

Los pasos 1 a 3 se repetirán veces de precisión (por defecto: 5) y las respuestas se promediarán utilizando el análisis rDiff.

Lo anterior forma parte de la creación de perfiles, pero Arachni también nos ofrece crear usuarios, con los que se puede compartir los resultados de los análisis, esto para que los equipos de trabajo puedan trabajar en conjunto.

Otras de sus características más resaltadas es que permite importar reportes de escáneres y se pueden descargar los documentos de reporte que se crearon con arachni, sea de tipo HTML, JSON, XML, entre otros.

21. Pruebas de funcionamiento con DVWA

Es momento de probar el funcionamiento de Arachni e identificar si todas sus funciones trabajan de manera fiable y cumplen con los requisitos que se necesitan para desempeñar el trabajo que requieren las aplicaciones que desarrolla el área de Software Factory.

A continuación se muestran algunas pruebas realizadas con una aplicación muy popular en internet, que es utilizada generalmente para aprender sobre las vulnerabilidades existentes y como pueden dañar la información del sitio, esta herramienta es DVWA (Damn Vulnerable Web Application).

DVWA es una aplicación hecha en PHP y MySQL para el entrenamiento de explotación de vulnerabilidades web**,** perfecta para poner a prueba las habilidades en el tema e igualmente para aprender nuevas técnicas. DVWA está divido en tres niveles: Low, medium y hight, cada uno respectivamente va aumentando su nivel de dificultad. Esta herramienta se instala y se prosigue haciendo el análisis de forma local con arachni.

DVWA permite el análisis del código vulnerable a los siguientes ataques:

- Fuerza Bruta
- Ejecución de Comandos
- CSRF (Cross-Site Request Forgery)
- Inclusión de Archivos (Inclusión de Archivos Locales e Inclusión de Archivos Remotos)
- Inyección SQL
- Inyección SQL (Ciega)
- Carga de Archivos
- XSS Reflejado
- XSS Almacenado

Primeramente se configuro DVWA utilizando XAMPP y se creó la base de datos que automáticamente se configura con la ejecución de la aplicación, agregando las tablas que se necesitan.

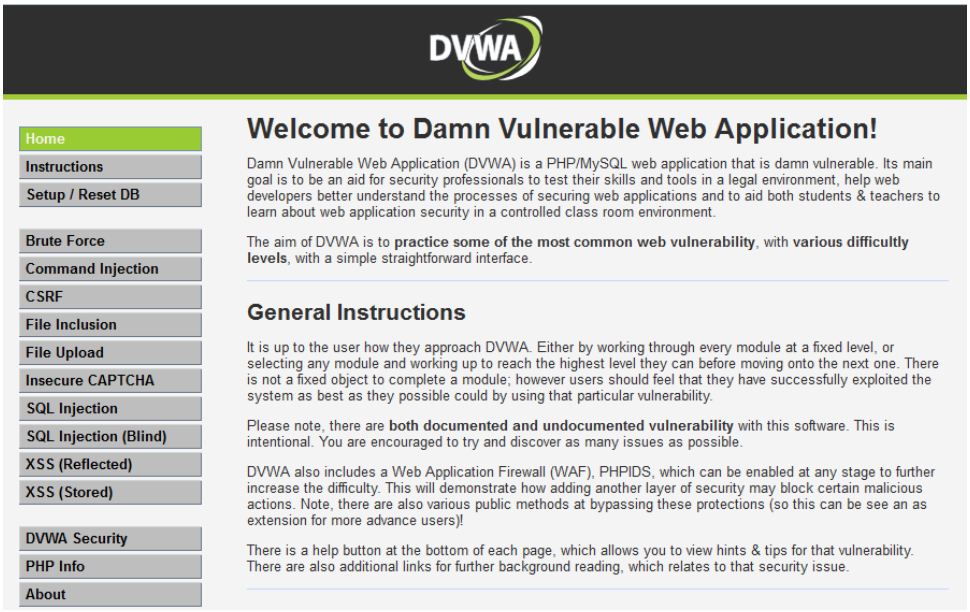

Ilustración 23. Pantalla de inicio de DVWA

Se harán algunas pruebas con DVWA asegurándonos de que efectivamente las vulnerabilidades que se mencionan en su descripción están implementadas.

Inyección SQL

Ilustración 24. Inyección SQL en DVWA

Inyección ciega SQL

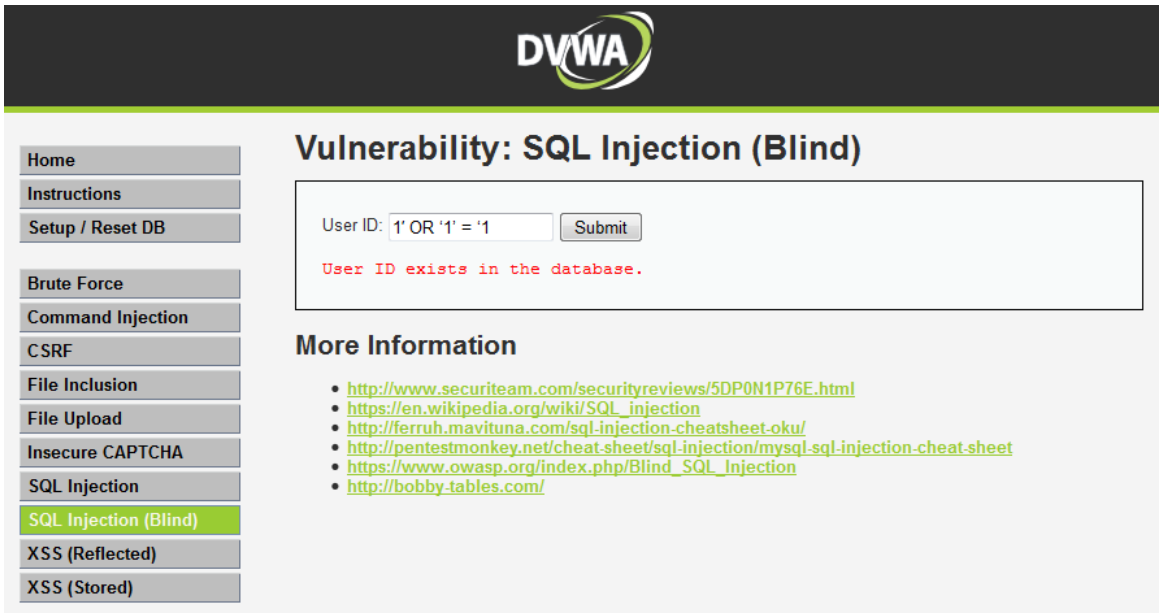

Ilustración 25. Inyección Ciega SQL en DVWA

XSS (Reflejado)

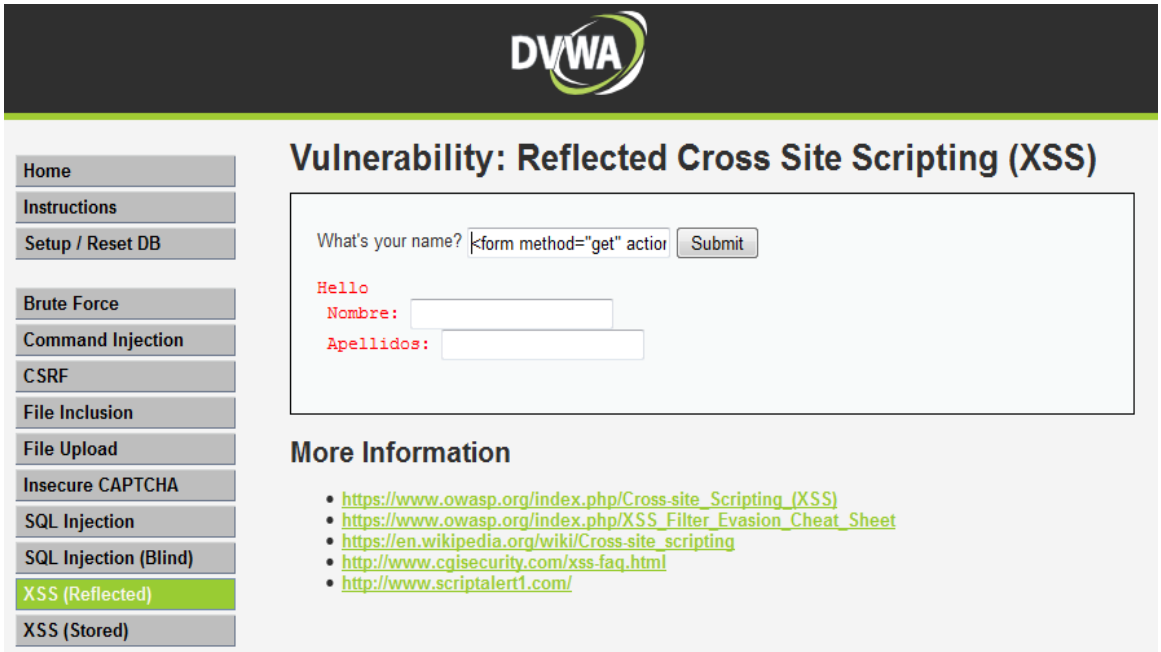

Ilustración 26. Ataque XSS en DVWA

Inclusión de Archivos Locales

Ilustración 27. Inclusión de Archivos Locales en DVWA

Ya con algunos ejemplos demostrados de la función de la aplicación que se va a analizar, se procede a llevar a cabo el escáner de Arachni. Cabe mencionar que esta prueba se está implementado para deliberar si Arachni es una aplicación competitiva y asegurar que funciona en un cien por ciento bien.

Al hacer el análisis con Arachni mostro resultados muy acertados, incluyendo algunos que no se mencionan, pero que solo son de información y recomendaciones.

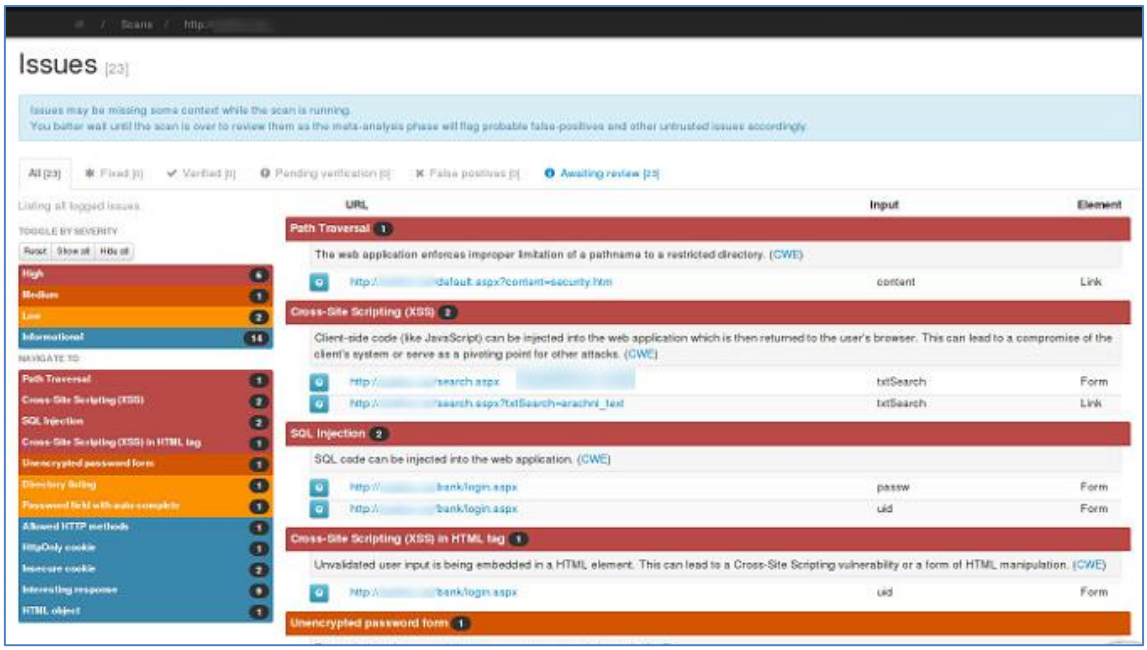

Ilustración 28. Resultados del escáner de DVWA

Arachni funciona correctamente y ofrece la capacidad de identificar por colores el grado de la vulnerabilidad, lo que es más fácil para el usuario. También muestra una pequeña descripción del problema encontrado y cuál es el impacto que tiene, asimismo muestra la lista de URL´s donde se presentó el problema.

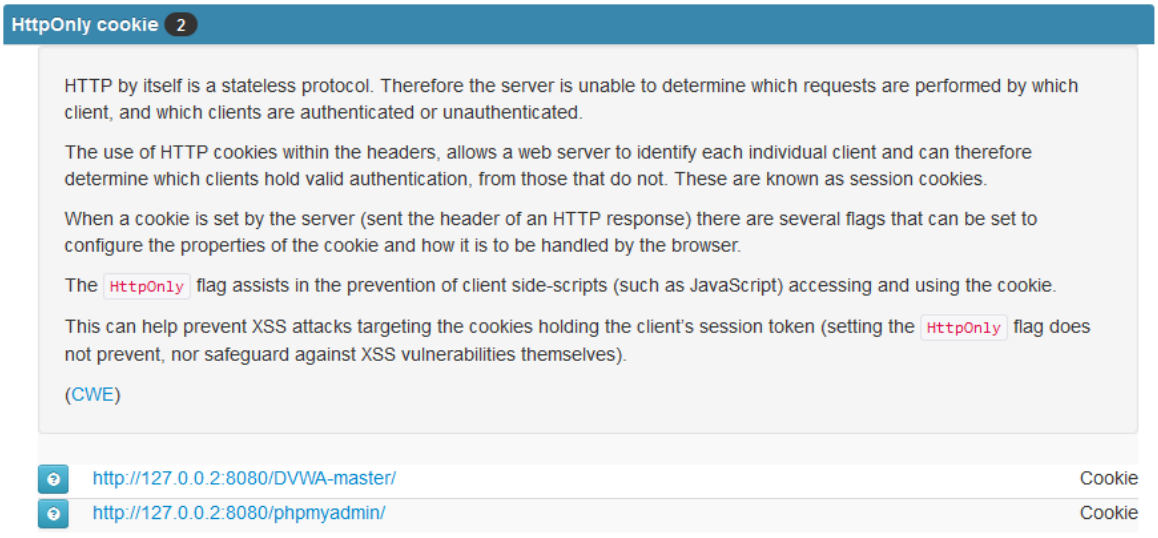

Ilustración 29. Detalles de descripción de vulnerabilidades

Al final del escáner se puede visualizar junto con el grupo de trabajo con el que se compartió.

Con esta prueba se puede corroborar que Arachni funciona correctamente, que su análisis es muy acertado y que es una herramienta muy competente y confiable.

22. Reporte de seguridad con aplicaciones de Software Factory

Después de haber probado el funcionamiento de Arachni con una aplicación de la que se conocía el resultado, es momento de hacer pruebas directas con las aplicaciones que desarrolla Software Factory.

Para esto se tomarán como base dos aplicaciones que ya están liberadas pero que tienen la misma estructura que las que desarrollan actualmente, esto para conocer el grado de vulnerabilidad que se ha dejado pasar.

A continuación solo se mostrará el resultado que devolvió Arachni para cada aplicación.

BMUMS

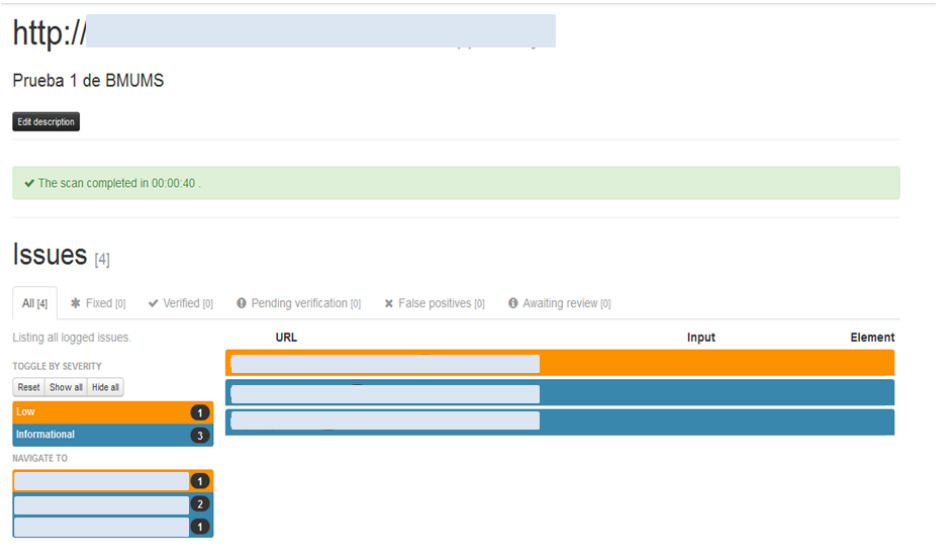

Ilustración 30. Prueba con BMUMS

SGT

Ilustración 31. Prueba con SGT

Ahora ya se podrá usar Arachni como herramienta de seguridad para las aplicaciones, es muy fácil y funciona adecuadamente.

Bibliografía

Alvarez, M. A. (2003). *desarrolloweb.com*. Recuperado el 2016, de http://www.desarrolloweb.com/articulos/1325.php

Buyto. (2009). *buyto.es*. Recuperado el 2016, de http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaciones-desktop

Cavsi. (2009). *cavsi.com*. Recuperado el 2016, de http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/

Diaz, R. (09 de 07 de 2014). *LinkedIn*. Recuperado el 22 de 02 de 2017, de https://www.linkedin.com/pulse/20140709164232-6227054-los-b%C3%A1sicos-de-la-administraci%C3%B3n-%C3%A1gil-de-proyectos-scrum

Gallego, M. T. (2013). *openaccess*. Recuperado el 2017, de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf

masadelante. (2004). *masadelante.com*. Recuperado el 2016, de http://www.masadelante.com/faqs/base-de-datos

microsoft. (2010). *msdn.microsoft.com*. Recuperado el 2016, de https://msdn.microsoft.com/es-MX/library/z1zx9t92.aspx

Nacional, I. P. (s.f.). *sites.upiicsa.ipn.mx*. Recuperado el 2016, de http://www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/p_terminados/PolilibroFC/Unidad_III/Unidad%20III_4.htm

Oracle. (2012). *www.java.com*. Recuperado el 2016, de https://www.java.com/es/download/faq/whatis_java.xml

Palacio, J. (2014). *scrummanager*. Recuperado el 2017, de http://www.scrummanager.net/files/sm_proyecto.pdf

Romaniz, S. C. (2015). *sedici.unlp.* Recuperado el 2017, de http://sedici.unlp.edu.ar/bitstream/handle/10915/21581/1927+-+Seguridad+de+aplicaciones+web+vulnerabilidades+en+los+controles+de+acceso.pdf;jsessionid=C9F64B36ACE5D5515D46F422F17CF14C?sequence=1

santos, B. D. (2005). *ecured.cu*. Recuperado el 2016, de https://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos

VASS digital. (2012). *vassdigital.com*. Recuperado el 2016, de http://www.vassdigital.com/scrum-la-metodologia-de-desarrollo-agil-por-excelencia/