**ANEXO II.**

Plan de trabajo propuesto por el alumno con el aval del Profesor Orientador

Formulario 1. Presentación de la Propuesta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre del Trabajo | Aplicación de medidas de seguridad contra XSS en un sitio web para el INICNE | | | |
| Consignar tres palabras claves de identificación del trabajo | Seguridad, Frameworks,MVC, XP | | | |
| Prof. Orientador (1) | Lic. Juan Francisco Bosco | | Dirección, teléfono y e-mail  boscojfrancisco@gmail.com | |
| Prof. Orientador (2) | Expt. Cuzziol, Juan José | | Dirección, teléfono y e-mail  jcuzziol@hotmail.com | |
| Apellido y Nombres del Alumno:  Silva, Pablo Ariel | Nro. de libreta  47062 | DNI  34825391 | | Dirección, teléfono y e-mail  Pabloariel01@hotmail.com |
| Clasificación del trabajo: | Desarrollo de aplicaciones/sistemas con la descripción de la metodología seguida en cada una de las etapas: Especificación de requerimientos, diseño, desarrollo. | | | |
| Área Informática | Seguridad, Ingeniería del software | | | |
| Cronograma | Ver Tabla 1 | | | |
| Trabajo enmarcado en | |  | | --- | | Institución: INICNE | | | | |

Cronograma

Tabla 1 - Cronograma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre de tarea | Duración | Comienzo | Fin | Predecesoras |
| Búsqueda de bibliografía | 1 día | mar 08/03/16 | mar 08/03/16 |  |
| **Primeros Pasos** | **20 días** | **mar 08/03/16** | **lun 04/04/16** |  |
| Elección del Tema | 5 días | mar 08/03/16 | lun 14/03/16 |  |
| Selección de Metodología | 5 días | mar 15/03/16 | lun 21/03/16 | 3 |
| Recolección de Información | 10 días | mar 15/03/16 | lun 28/03/16 | 3 |
| Búsqueda del Profesor Orientador | 5 días | mar 15/03/16 | lun 21/03/16 | 3 |
| Desarrollo de Primera Entrega | 5 días | mar 29/03/16 | lun 04/04/16 | 5 |
| **Primer entrega de borrador** | **15 días** | **mar 05/04/16** | **lun 25/04/16** | **7** |
| Entrega del primer borrador | 1 día | mar 05/04/16 | mar 05/04/16 | 7 |
| Devolución de la Primer Entrega | 7 días | mié 06/04/16 | jue 14/04/16 | 9 |
| Correcciones de la entrega | 7 días | vie 15/04/16 | lun 25/04/16 | 10 |
| **Entrega del Segundo Borrador** | **15 días** | **mar 26/04/16** | **lun 16/05/16** |  |
| Entrega del Borrador | 1 día | mar 26/04/16 | mar 26/04/16 | 11 |
| Devolución de la Entrega | 7 días | mié 27/04/16 | jue 05/05/16 | 13 |
| Correcciones de la Entrega | 7 días | vie 06/05/16 | lun 16/05/16 | 14 |
| **Entrega del Tercer Borrador** | **15 días** | **lun 01/08/16** | **vie 19/08/16** |  |
| Entrega del Borrador | 1 día | lun 01/08/16 | lun 01/08/16 | 12 |
| Devolución de la Entrega | 7 días | lun 01/08/16 | mar 09/08/16 | 17 |
| Correcciones de la Entrega | 7 días | lun 01/08/16 | mar 09/08/16 | 18 |
| **Desarrollo de la aplicación** | **125 días** | **vie 15/04/16** | **jue 06/10/16** |  |
| **Primera Iteración** | **55 días** | **vie 15/04/16** | **jue 30/06/16** |  |
| Análisis | 20 días | vie 15/04/16 | jue 12/05/16 |  |
| Diseño | 20 días | vie 13/05/16 | jue 09/06/16 | 22 |
| Codificación y Pruebas | 15 días | vie 10/06/16 | jue 30/06/16 | 23 |
| **Segunda Iteración** | **35 días** | **vie 01/07/16** | **jue 18/08/16** | **21** |
| Análisis | 10 días | vie 01/07/16 | jue 14/07/16 |  |
| Diseño | 10 días | vie 15/07/16 | jue 28/07/16 | 26 |
| Codificación y Pruebas | 15 días | vie 29/07/16 | jue 18/08/16 | 27 |
| **Tercera Iteración** | **35 días** | **vie 19/08/16** | **jue 06/10/16** | **25** |
| Análisis | 10 días | vie 19/08/16 | jue 01/09/16 |  |
| Diseño | 10 días | vie 02/09/16 | jue 15/09/16 | 30 |
| Codificación y Pruebas | 15 días | vie 16/09/16 | jue 06/10/16 | 31 |

1. Introducción

* Breve estado del arte

Un tiempo atrás existía el inconveniente de que para proporcionar a los usuarios una aplicación con muchas funcionalidades, esta debía ser de escritorio y había que tener en cuenta el trabajo que llevaba codificar más instalar el aplicativo en cada una de las computadoras que lo van a utilizar.

En la actualidad poseemos más opciones, AJAX [1] permite desarrollar una aplicaciones basadas en el navegador, que tenga todas las funcionalidades requeridas y de fácil acceso, solo precisan de un navegador web para ser utilizadas.

El uso de esta técnica también tiene sus inconvenientes, entre ellos nuevos riesgos de seguridad, al exponer una aplicación a la web se incrementa el número de personas que puede acceder a la misma, y nada garantiza que una de esas personas intente atentar contra la aplicación y un ataque a un sitio web vulnerable podría dejar expuesta información sensible o privada.

Actualmente existe un proyecto llamado OWASP [2] que realiza periódicamente un TOP 10 de los principales riegos de las aplicaciones web, entre los que se encuentran las inyecciones [3] y las secuencias de comandos de sitios cruzados como los más explotados.

En este trabajo se aplicaran medidas para mitigar los ataques XSS [4] [5] [6] [7] por lo que se controlarán las entradas a las funciones de la aplicación y las salidas de las mismas para prevenir que estas causen un comportamiento indebido [6].

* Objetivos

Objetivo general:

Estudiar algunas técnicas de Seguridad Web orientadas a la prevención de inyecciones de código malicioso y aplicarlas en el desarrollo de un sistema Web para el INICNE.

Objetivos específicos:

* Profundizar el estudio de conceptos de seguridad web
* Aplicar medidas de protección contra Cross Site scripting(XSS).
* Diseñar y desarrollar una base de datos.
* Implementar un framework para aumentar la seguridad, crear una aplicación estructurada, y con código legible y mantenible.
* Fundamentación

Muchas veces se confía que las aplicaciones web donde se deposita información son confiables y que los datos están seguros. Según [8] se puede observar que la mayoría de las bases de datos de las organizaciones carecen de seguridad suficiente o se desconoce si existen medidas de seguridad y se debe considerar que se puede llegar a una base de datos a través de su aplicación si se poseen los conocimientos necesarios.

En este trabajo se estudiarán distintas formas de protegerse contra las secuencias de comandos de sitios cruzados (XSS) los cuales son riesgos presentes desde ya hace muchos años. Se puede ver en las encuestas de WhiteHat [9] que las consecuencias de que se explote una vulnerabilidad de este tipo podría tener efectos negativos muy importantes para los usuarios, sin embargo muchas de estas vulnerabilidades siguen estando activas, gracias a sistemas heredados en los cuales agregar módulos nuevos resulta muy costoso, o aplicaciones modernas con fallas en seguridad [10].

Para llevar a la práctica estos conocimientos, se desarrollara una aplicación multiusuario con orientación web para el INICNE (Instituto de Ictiología del Nordeste), que actualmente carece de un sistema, utiliza planillas Excel individuales como medio de resguardo de la información tanto para su archivado como para su análisis.

1. Metodología

* Ciclo de vida del proyecto

Se optará por el uso de XP [11] [12] sobre el cual se realizarán cambios. Al ser un proyecto individual la programación de a pares se ve dificultada, por lo que será reemplazada por la revisión de código por parte del profesor orientador. La etapa de análisis de cada una de las iteraciones se verá beneficiada con el agregado que se pueda obtener de un breve análisis de ingeniería inversa [13] realizado sobre las planillas Excel existentes, las que también servirán como criterios de aceptación y para corroborar que la aplicación funciona correctamente. También se modelarán amenazas de aplicación basado en las recomendaciones de OWASP [14].

En la etapa de desarrollo se seguirá una metodología basada en [15], consistiendo en pasar las entradas por distintos filtros, incluyendo librerías para escapar caracteres que puedan ser peligrosos y limpiar el código ingresado, y por último se usaran técnicas de codificación para prevenir ataques en entornos HTML.

* Plan de entregas

XP propone entregas continuas y pruebas con el cliente, para lo cual hay que realizar una planificación previa en base a valoraciones que hace el cliente respecto a que aspectos del sistema los considera críticos para su negocio. En este proyecto se propone realizar dos entregas, y segmentarlas en tres Sprints, el primero será un sprint 0, útil para sentar bases, pero no funcional, y los dos siguientes Sprints serán entregas de módulos de la aplicación, representativas del trabajo.

1. Ámbito de trabajo

El trabajo se desarrollará principalmente de forma independiente, y se trabajará consultando a los futuros usuarios de la aplicación con cliente, en vez de con cliente ín situ como propone la metodología XP, y acudiendo a la FaCENA para consultas a los profesores orientadores y profesores de la catedra.

1. Recursos

Como recursos informáticos se hará uso de una computadora en la cual se programará con HTML, PHP y JavaScript con el uso un framework tanto para PHP como para Javascript, ejecutados sobre un servidor Apache y abasteciéndose de una base de datos MySQL.

Para realizar la migración de Excel a MySQL se utilizará la herramienta Pentaho.

Para realizar esto se hará uso de las documentaciones de los lenguajes frameworks, como también uso de bibliografía adicional para profundizar en la temática de la seguridad.

A demás se utilizará Sublime Text como editor de texto, Git versionar la aplicación y Mozilla Firefox como navegador para realizar las pruebas.

Por otro lado también serán necesarios recursos humanos, como la persona que llevará a cabo el trabajo, y un profesor orientador

1. Desarrollos propuestos

En este trabajo se propone investigar y aplicar medidas defensivas y preventivas para proteger los datos y la aplicación, empezando por la protección contra accesos indebidos, y realizando acciones para mitigar intentos de inyección maliciosa de código o Cross Site Scripting.

La aplicación que se propone diseñar y desarrollar será web, capaz de almacenar y explotar grandes volúmenes de datos muestrales, para lo cual se configurará un servidor Apache, para hospedar la aplicación y actuar como una medida más de seguridad, evitando accesos indebidos y registrando distintos tipos de sucesos.

Dentro de la aplicación se usaran distintos perfiles de usuarios, aquellos dados de alta por un administrador podrán acceder a la aplicación, todos podrán consultar datos y generar reportes, y se asignarán permisos dependiendo del tipo de usuario. El usuario básico solo podrá realizar lecturas, un usuario será destinado a la carga de datos y el administrador tendrá la capacidad de gestionar las cuentas de usuario y realizar altas y modificaciones y resguardos.

Por otra parte se diseñará e implementará una base de datos. La misma se creará en base a datos tabulados en planillas Excel que contienen actualmente los datos.

Como un adicional se realizara el proceso ETL para migrar de las planillas Excel a la base de datos

1. Resultados esperados

Se espera aprender sobre los distintos riesgos de seguridad de aplicaciones web, especialmente sobre Cross Site Scripting, ya que para defenderse se debe conocerlos.

También se diseñará una aplicación web que permita satisfacer las necesidades del INICNE, y una guía para futuros alumnos que quieran profundizar más en la programación web, ya sea con el uso de Frameworks en entornos web, como también en realizar aplicaciones más seguras.

1. Referencias bibliográficas

* Referencias

|  |  |
| --- | --- |
| [1] | R. Asleson y N. T. Schutta, Foundation of AJAX, apress, 2005. |
| [2] | OWASP, «OWASP Top Ten Project,» 12 junio 2013. [En línea]. Available: www.owasp.org/images/5/5f/OWASP\_Top\_10\_-\_2013\_Final\_-\_Espa%C3%B1ol.pdf. |
| [3] | E. Couture, «Web Application Injection Vulnerabilities a Web App’s Security Nemesis?,» GIAC (GWAPT) Gold Certification, SANS Institute, Mayo 2013. |
| [4] | B. B. Guptaa, K. P. Meenaa, S. Guptaa, S. Gangwara y M. Kumara, «Cross-Site Scripting (XSS) Abuse and Defense: Exploitation on Several Testing Bed Environments and Its Defense,» *Journal of Information Privacy and Security,* vol. 11, nº 2, pp. 118-136, 2015. |
| [5] | P. Brady, «Survive the Deep End: PHP Security,» 26 Octubre 2015. [En línea]. Available: http://phpsecurity.readthedocs.org/en/latest/. |
| [6] | D. Stuttard y M. Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Indianapolis, Indiana: John Wiley & Sons, Inc., 2011. |
| [7] | S. Forgie, J. Grossman, R. Hansen, A. Rager y P. D. Petkov, XSS Attacks: Cross Site Scripting Exploits and Defense., Burlington, MA: Syngress Publishing, Inc., 2007. |
| [8] | J. McKendrick, «Dbs. Security Superhero: IOUG Enterprise Data Security Survey,» Unisphere Research, New Jersey, 2014. |
| [9] | WhiteHat Security, « WhiteHat Security Status Report,» 2015. [En línea]. Available: https://info.whitehatsec.com/rs/whitehatsecurity/images/2015-Stats-Report.pdf. |
| [10] | L. Salgado, . M. Ron y F. Solis, «Análisis de Riesgos de las Aplicaciones Web de la Superintendencia de Bancos y Seguros, Utilizando las Recomendaciones TOP TEN de OWASP,» Universidad de las Fuerzas Armadas ESPE, 2014. |
| [11] | D. Wells, «Extreme Programming,» 8 october 2013. [En línea]. Available: http://www.extremeprogramming.org/. [Último acceso: 10 junio 2016]. |
| [12] | K. Beck y C. Andres, Extreme Programming Explained: Embrace Change, Addison-Wesley, 2004. |
| [13] | R. S. Pressman, «Modelos de Proceso Prescriptivo,» de *Ingeniería del Software: un enfoque práctico*, New York, The McGrawl-Hill Company, inc, 2010, pp. 33-42. |
| [14] | OWASP, «Application Threat Modeling,» 8 Marzo 2015. [En línea]. Available: https://www.owasp.org/index.php/Application\_Threat\_Modeling. [Último acceso: 10 Junio 2016]. |
| [15] | «XSS (Cross Site Scripting) Prevention Cheat Sheet,» OWASP, 27 marzo 2016. [En línea]. Available: https://www.owasp.org/index.php/XSS\_(Cross\_Site\_Scripting)\_Prevention\_Cheat\_Sheet. [Último acceso: 15 mayo 2016]. |