**ANEXO II.**

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

# Formulario 1. Presentación de la Propuesta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre del Trabajo |  | | | |
| Consignar tres palabras claves de identificación del trabajo | XSS, Seguridad | | | |
| Prof. Orientador (1) | Lic. Juan Francisco Bosco | | Dirección, teléfono y e-mail | |
| Prof. Orientador (2) |  | | Dirección, teléfono y e-mail | |
| Apellido y Nombres del Alumno:  Silva, Pablo Ariel | Nro. de libreta  47062 | DNI  34825391 | | Dirección, teléfono y e-mail |
| Clasificación del trabajo:  (s/ Anexo I, pto.3) |  | | | |
| 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** | **7 días** | **lun 01/08/16** | **mar 09/08/16** |  |
| Entrega del Borrador | 1 día | lun 01/08/16 | lun 01/08/16 |  |
| Devolución de la Entrega | 7 días | lun 01/08/16 | mar 09/08/16 |  |
| Correcciones de la Entrega | 7 días | lun 01/08/16 | mar 09/08/16 |  |
| **Desarrollo de la aplicación** | **125 días** | **vie 15/04/16** | **jue 06/10/16** |  |
| **Primera Iteracion** | **55 días** | **vie 15/04/16** | **jue 30/06/16** |  |
| Analisis | 20 días | vie 15/04/16 | jue 12/05/16 |  |
| Diseño | 20 días | vie 13/05/16 | jue 09/06/16 | 22 |
| Codificacion y Pruebas | 15 días | vie 10/06/16 | jue 30/06/16 | 23 |
| **Segunda Iteracion** | **35 días** | **vie 01/07/16** | **jue 18/08/16** | **21** |
| Analisis | 10 días | vie 01/07/16 | jue 14/07/16 |  |
| Diseño | 10 días | vie 15/07/16 | jue 28/07/16 | 26 |
| Codificacion y Pruebas | 15 días | vie 29/07/16 | jue 18/08/16 | 27 |
| **Tercera Iteracion** | **35 días** | **vie 19/08/16** | **jue 06/10/16** | **25** |
| Analisis | 10 días | vie 19/08/16 | jue 01/09/16 |  |
| Diseño | 10 días | vie 02/09/16 | jue 15/09/16 | 30 |
| Codificacion y Pruebas | 15 días | vie 16/09/16 | jue 06/10/16 | 31 |

1. Introducción
   * Breve estado del arte

Un tiempo atrás existía el inconvenite 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 mas instalar el aplicativo en cada una de las computadoras que lo van a utilizar.

En la actualidad poseemos mas opciónes, AJAX [1] nos 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 reiesgos de seguridad, al exponer una aplicación a la web se incrementa el numero de personas que puede acceder a la misma, y nada nos garantiza que una de esas personas intente atentar contra la aplicación y un ataque a un sitio web vulnerable podría dejar expuesta informacion 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 van a aplicar medidas para mitigar los ataques XSS [4] [5] [6] [7] para lo que se hará será controlar 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 las 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:

-Aplicar medidas de protección contra Cross Site scripting.

-profundizar el estudio de conceptos de seguridad web.

-Aplicar medidas del lado del servidor web para aumentar la seguridad de la aplicación.

-Implementar un framework para aumentar la seguridad, crear una aplicación estructurada, y con un código legible y mantenible.

-Diseñar y desarrollar una base de datos.

* + Fundamentación

Muchas veces se confía que las aplicaciones web donde se deposita nuestra información es confiable 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 travez de su aplicación si se poseen los conocimientos necesarios.

En este trabajo se estudiaran distintas formas de protegerse contra las secuencias de comandos de sitios cruzados (XSS) los cuales a pesar de que son riesgos desde ya hace muchos años, se puede ver en las encuestas de WhiteHat [9] debido a que se considera que los efectos de sufrir un ataque de este tipo podría tener efectos negativos muy importantes para los usuarios, y siguen estando activas, gracias a vulnerabilidades en sistemas heredados, 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.

Con una aplicación web multiusuario y una base de datos centralizada se disminuiría drásticamente la taza de errores al compartir información, se bajarían los tiempos de armado de tablas de una semana a un par de clics y se podrían obtener datos estadísticos de forma automática y simple.

1. Metodología
2. Ciclo de vida del proyecto

Se optara por el uso de XP [11] [12] sobre el cual se realizaran cambios. Al ser un proyecto individual no se realizara programación de a pares. A demás la etapa de análisis de cada una de las iteraciones se vera beneficiada con el agregado que se pueda obtener de un breve análisis de ingeniería inversa [11] realizado sobre las planillas Excel existentes, las que también servirán para corroborar que la aplicación funciona correctamente.

En la tapa de análisis también se anadira un análisis de riesgos basado en [dread/stride] y en la etapa de desarrollo se seguirá una metodología basada en [12], consistiendo en pasar las entradas por 3 filtros, el primero para escapar caracteres que puedan ser peligrosos, el segundo consiste en usar librerías para limpiar el código ingresado y por último se usaran técnicas de codificación para prevenir ataques en entornos HTML.

1. Ámbito de trabajo

El trabajo se desarrollara principalmente de forma independiente, y se trabajara 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

1. Recursos

Como recursos informáticos hará uso de una computadora en la cual se programara 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 utilizara la herramienta Pentaho.

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

A demás se utilizara Sublime Text como editor de texto, Git llevar a cabo un versionado de la aplicación y Mozilla Firefox como navegador para realizar las pruebas.

Por otro lado también serán necesarios recursos humanos, como la persona que llevara 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, siguiendo con la administración de la base de datos para anular posibles intentos de operaciones indebidas y finalmente, 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 configurara un servidor Apache, no solo para que pueda hospedar la aplicación sino para que pueda actuar como una medida más de seguridad, evitando accesos indebidos y registrando distintos tipos de sucesos.

Dentro de la a aplicación se usaran distintos perfiles de usuarios, solo usuarios dados de alta por un administrador podrán acceder a la aplicación, todos los usuarios podrán consultar datos y generar reportes, y el administrador tendrá la capacidad de gestionar las cuentas y será el único con la capacidad de realizar altas y modificaciones.

Por otra parte se deberá diseñar e implementar una base de datos. La misma se deberá crear 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 poder aprender sobre los distintos riesgos de seguridad de aplicaciones web, especialmente sobre Cross Site scripting, ya que para poder defenderse se debe conocer cuáles son los posibles riesgos.

También espera diseñar una aplicación que pueda servir no solo para satisfacer las necesidades del INICNE, sino también como una guía para futuros alumnos que quieran profundizar más en la programación web, ya sea con el uso de Frameworks para php, como también en realizar aplicaciones más seguras.

1. Referencias bibliográficas

# Referencias

|  |  |
| --- | --- |
| [1] | 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. |
| [2] | A. Bamrara, «Evaluating Database Security and Cyber Attacks: A Relational Approach,» *Journal of Internet Banking and Commerce,* vol. 20, nº 2, 2015. |
| [3] | E. Couture, «Web Application Injection Vulnerabilities a Web App’s Security Nemesis?,» GIAC (GWAPT) Gold Certification, SANS Institute, Mayo 2013. |
| [4] | J. McKendrick, «Dbs. Security Superhero: IOUG Enterprise Data Security Survey,» Unisphere Research, New Jersey, 2014. |
| [5] | P. Brady, «Survive the Deep End: PHP Security,» 26 Octubre 2015. [En línea]. Available: http://phpsecurity.readthedocs.org/en/latest/. |
| [6] | B. Gupta, «Web Application Security – What You Need to Know,» de *Thirty-Third Annual Pacific Northwest Software Quality Conference*, World Trade Center Portland Portland, Oregon. pp 271-280, Octubre 2015. |
| [7] | WhiteHat Security, « WhiteHat Security Status Report,» 2015. [En línea]. Available: https://info.whitehatsec.com/rs/whitehatsecurity/images/2015-Stats-Report.pdf. |
| [8] | 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. |
| [9] | D. Stuttard y M. Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Indianapolis, Indiana: John Wiley & Sons, Inc., 2011. |
| [10] | 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. |
| [11] | D. Hazaël-Massieux, «W3C,» [En línea]. Available: https://www.w3.org/standards/webdesign/script. |
| [12] | H. W. Lie y B. Bos, «W3C,» 11 4 2008. [En línea]. Available: https://www.w3.org/TR/2008/REC-CSS1-20080411/. |
| [13] | W3C, «W3C: HTML & CSS,» [En línea]. Available: https://www.w3.org/standards/webdesign/htmlcss. |
| [14] | . T. O'Reilly, « What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software.,» *Communications & Strategies,* nº 65, pp. 17-37, 2007. |
| [15] | A. T. Holdener III, Ajax: The Definitive Guide, North, Sebastopol, CA: O'Reilly Media, 2008. |
| [16] | M. Heiderich, E. A. Vela Nava, G. Heyes y D. Lindsa, Web Application Obfuscation ‘-/WAFs..Evasion..Filters//alert(/Obfuscation/)-’, Burlington, MA: Elsevier, 2011. |
| [17] | A. MacCaw, JavaScript Web Applications, North, Sebastopol, CA: O'Reilly Media, 2011. |