Seguridad en redes: ¿Qué es? ¿Cómo lograrla?

Gunnar Wolf — gwolf@gwolf.org
http://www.gwolf.org/seguridad/seg_en_redes

Instituto de Investigaciones Económicas, UNAM
Miembro externo del Departamento de Seguridad en Cómputo DGSCA-UNAM
Desarrollador del proyecto Debian

FLISOL Monterrey, Universidad Regiomontana
28 de febrero, 2008

debian

Contenidos

debian

¿Qué es la seguridad?

A ver, ¿quién se atreve a responder?

- Que el sistema se comporte como esperamos
- ummm...
  Que el sistema se comporte como *nosotros* (sus legítimos programadores, administradores y usuarios del sistema) esperamos

Suena un poco vago, ¿no? Bueno, vamos un poco más a detalle...

debian

¿Qué es la seguridad?

A ver, ¿quién se atreve a responder?

- Que el sistema se comporte como esperamos
- ummm...
  Que el sistema se comporte como *nosotros* (sus legítimos programadores, administradores y usuarios del sistema) esperamos

Suena un poco vago, ¿no? Bueno, vamos un poco más a detalle...

debian

¿Qué es la seguridad?

A ver, ¿quién se atreve a responder?

- Que el sistema se comporte como esperamos
- ummm...
  Que el sistema se comporte como *nosotros* (sus legítimos programadores, administradores y usuarios del sistema) esperamos

Suena un poco vago, ¿no? Bueno, vamos un poco más a detalle...

debian

Pero antes... La palabra mágica

Antes de continuar, tenemos que comprender una palabra muy importante.
Una palabra mágica.

Adecuado

Todo lo que mencione a partir de este punto tiene que ir matizado con esta palabra.

debian

Pero antes... La palabra mágica

Antes de continuar, tenemos que comprender una palabra muy importante.
Una palabra mágica.

Adecuado

Todo lo que mencione a partir de este punto tiene que ir matizado con esta palabra.

debian

Características de un sistema seguro

¿Qué características tiene un sistema seguro?

Dicho sea de otro modo...

¿Cómo puedo atreverme a afirmar que un sistema es *suficientemente seguro* para mis necesidades?

debian

Consistencia

- Ante las mismas circunstancias, el sistema debe presentar el mismo comportamiento

debian

Protección y separación

- Los datos, instrucciones y espacio de memoria de un programa no deben interferir ni ser interferidos por otros
- Los datos y las acciones de un usuario no deben ser visibles o modificables por otros, y no deben tener efecto para otros.
- Las condiciones anormales de un proceso —sean accidentales o expresas— deben tener un impacto mínimo en el sistema

debian

Autenticación

- El sistema debe poseer los mecanismos necesarios para asegurarse que un usuario es realmente quien dice ser

debian

Control de acceso / Autorización

- El administrador del sistema, así como cada usuario, deben poder controlar granularmente los permisos de acceso a su información — Quién tiene acceso a qué recursos, y qué tipo de acceso tiene.
- El sistema debe asegurarse de que los usuarios no se *brinquen* estas restricciones

Auditoría

- El sistema debe ser capaz de registrar, así como de notificar al administrador (y opcionalmente a los usuarios), de cualquier anomalía o evento importante

¿Qué significa *importante*? ¿Cómo puede saberlo mi computadora?

debian

Auditoría

- El sistema debe ser capaz de registrar, así como de notificar al administrador (y opcionalmente a los usuarios), de cualquier anomalía o evento importante

¿Qué significa *importante*? ¿Cómo puede saberlo mi computadora?

debian

El utópico 100 %

- Es imposible alcanzar un 100 % de seguridad.
- Los programas son escritos por humanos, y por tanto son susceptibles a tener todo tipo de errores
- La complejidad de los programas -aún los más simples- no permite al programador mantener en mente todos los factores
- Hay una gran cantidad de interacciones entre los elementos de un programa y el sistema, y un cambio en cualquiera de ellos puede tener consecuencias inesperadas si no se hace con cuidado y conciencia
- Constantemente aparecen nuevas categorías de errores capaces de llevar a problemas de seguridad

Contenidos

debian

Capa física

- Insuficiente control de acceso al edificio
- Cableado mal instalado
  - Cableado no entubado
  - Puntos de red en lugares impensables
- Inalámbrica no cifrada (o demasiado débil)
- Facilidad de conexión sin trámites
- Mal manejo de la basura

debian

Sistemas operativos

- Sistemas poco tolerantes a situaciones frontera o a datos mal formados
- Revisión no exhaustiva de posibles respuestas
- Falta de una auditoria real y constante
- APIs/ABIs cambiantes
- Excesivo bagaje de compatibilidad *hacia abajo*

debian

Aplicaciones

- Confianza ciega en los datos recibidos
  - Longitud de los datos (por *buffers* y por truncado silencioso)
  - Tipo de los datos
  - Confianza en los datos originados externamente
- Uso de bibliotecas inestables
- Uso de *frameworks* no comprendidos suficientemente por el desarrollador
- Uso de protocolos de red obsoletos

Usuarios

El coco de los administradores...

- Políticas de uso aceptable inexistentes, incompletas, no aplicables o carentes de autoridad
- Falta de conciencia
- Capa 8 defectuosa
- Errores de BIOS
- La famosa *lucha entre los programadores y el universo*
- Analogías: Las contraseñas y la ropa interior

debian

Contenidos

debian

¿¡Ecología cibernética!?

¿Qué es la ecología?

La importancia de ser conscientes de nuestro entorno

...Y de actuar en consecuencia

Apliquemos eso a nuestro entorno cibernético

debian

Ser programadores seguros

- KISS: Keep It Simple, Stupid!
- DRY: Don't Repeat Yourself
- YAGNI: You Ain't Gonna Need It
- Pensar en los posibles futuros programadores/mantenedores de nuestro software
- Revisar siempre todo — Valores de retorno, resultado de la creación de objetos, tipos de datos, contenido válido, manejo de excepciones...
- **Desconfiar especialmente de los datos provenientes del usuario**
- Es más importante escribir un programa correctamente que clavarnos en detalles. *La optimización prematura es la fuente de todos los males*.

Ser programadores seguros

- Llevar a cabo un buen análisis previo a comenzar a escribir código
- Utilizar estructuras de datos comprensibles y acordes a nuestro problema
- Documentar conforme programamos
- Escribir pruebas conforme documentamos, basándonos en la documentación y no en nuestro otro código. De ser posible, que las pruebas las escriban equipos diferentes que quienes escribieron el código.
- Ofrecer a quien use nuestro código APIs consistentes, estables y bien documentadas

Ser administradores seguros

- Filosofía "Less is more": La belleza del minimalismo
- No hay nada como una sana paranoia
- Buscando la comodidad y la facilidad en el lugar correcto: Usar un OS bien diseñado y simple de administrar, *aunque tenga que aprender a utilizarlo*
- Caracterizar las verdaderas necesidades de los usuarios, saber decirles que no, ayudarles a ajustar lo que buscan.
- Mi usuario no es malicioso por naturaleza — aunque tal vez sí profundamente ignorante

debian

Ser usuarios seguros

- No confiar en que nadie atacará nuestro sistema únicamente por ser casero.
  - Recuerden a los virus, troyanos, gusanos y demás... Y a las hordas de script kiddies
  - Spambots y similares: No importan los recursos de *cada* sistema atacado; importa la *cantidad total* de sistemas sencillitos
- No restar importancia a las actualizaciones. (¿Sí tienes tus licencias en órden? Más te vale...)
- No restar importancia a la solidez general del sistema
- Enviar bug reports
- ...¿Realmente necesitamos un entorno de usuario tan rico? ¿Qué requerimos en realidad?

debian

Contenidos

debian

¿Qué es el Software Libre?

Es todo programa cuya licencia nos permite expresamente:

- Uso irrestricto, bajo cualquier circunstancia y para cualquier fin, del programa en cuestión
- Acceso irrestricto a su código fuente para aprender de él
- Derecho de adecuar, modificar, adaptar, corregir, mejorar su código
- Libre redistribución, gratuita o comercial, del programa completo o porciones del mismo, en su forma original o modificado

«libre» no es «gratis»

Es incorrecto referirse al Software Libre como software gratuito

- Es válido cobrar por desarrollar Software Libre
- Es válido cobrar por adecuar Software Libre
- Es válido cobrar por distribuir Software Libre

Claro, a quien se lo distribuyamos puede re-distribuirlo — Esto lleva a una autoregulación del mercado

debian

«libre» no es «dominio público» ni «shareware»

El Software Libre no es perteneciente al dominio público.

- El Software Libre está protegido por una licencia (copyright, copyleft) que impone claras restricciones y condiciones de uso/distribución
- El Software Libre pertenece a sus autores o a quien ellos designen

El Software Libre no es shareware

- No condiciona funcionalidad al pago de regalías
- Se distribuye con todo y fuentes
- No condiciona la distribución a un pago

debian

¿Por qué confiar en el Software Libre?

- El código está disponible — puede ser analizado por quien quiera (y sepa) hacerlo.
- Generalmente se apega estrictamente a estándares que permiten la operación en redes heterogéneas
- Muchas veces es desarrollado por interés personal, podríamos hasta decir que de forma altruista, no siguiendo los compromisos de negocios de una compañía
  - Como producto de investigación formal
  - Porque es divertido
  - Como ejercicio intelectual
  - «Ráscate donde te pique»

debian

¿Por qué confiar en el Software Libre?

- Muchos proyectos son resultado de colaboración de desarrolladores de todo el mundo
  - El código producido es más limpio y está mejor documentado que en desarrollos cerrados
  - El código mismo *es la única manera de comunicación* entre los desarrolladores — Eso exige muy altos niveles de control de calidad.
  - Es más extensible y adecuable, pues los desarrolladores comparten la filosofía de la libertad y porque están acostumbrados a aprender de otros proyectos libres
  - Hay menos «ceguera cultural» en los productos resultantes

debian

No volatilidad del contenido

- La razón de existir de todo nuestro software claramente es dar soporte al contenido
- Pensando a futuro, no podemos confiar la base de nuestra casa / negocio / empresa / gobierno /planeta en productos propietarios
  - Nuevos formatos incompatibles/falta de soporte para formatos de versiones anteriores
  - Dependencia de que un sólo fabricante continúe dando soporte al producto
  - Nos resta la capacidad de hacer cambios en nuestra tecnología base (hardware, OS, etc.)
  - El rol de los estándares oficiales en los formatos de archivos — Y el ridículo de ISO frente a OOXML

Programadores conscientes

- Hay más proyectos, más pequeños, menos complejos y que interoperan mejor entre sí que en los grandes productos comerciales
  - Es más fácil auditar código de proyectitos independientes que de grandes proyectos
  - Es más fácil reescribir componentes que no cumplan ciertos criterios
  - Es MUY IMPORTANTE mantener en el sistema una lista de dependencias y demás relaciones entre paquetes
- La seguridad es mayormente parte del inconsciente colectivo
- Otros programadores van a leer mi código, más vale que no haga cochinadas...

debian

Contenidos

debian

¿Dudas o comentarios?

¡Gracias!

Gunnar Wolf — gwolf@gwolf.org

http://www.gwolf.org/seguridad/seg_en_redes

debian