Curso: (62612) Diseño de aplicaciones seguras

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Introducción

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Un índice

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Introducción. Antes de empezar.

- Se invierte mucho tiempo, dinero y esfuerzo en seguridad a nivel de red por la mala calidad de los programas.
- Los antivirus, los cortafuegos, los sistemas de detección de intrusos (IDS) ayudan.
- Los programas malos son mucho más abundantes de lo que creemos.
- La forma de desarrollar los programas es responsable en gran medida del problema.

Cifras

- En 2002 el 'National Institute of Standards and Technology' (NIST) estimó que los defectos de los programas costaban mas de 60 millardos de dólares (60 billions).
  - Detectarlos a tiempo ahorraría 22 millardos de dólares.
    Citado en:
    'Measuring software quality. A Study of Open Source Software' Coverity, 2006.
    http://osvdb.org/ref/blog/open_source_quality_report.pdf
    http://www.coverity.com/library/pdf/open_source_quality_report.pdf

Cifras

- Menos del 10 % de proyectos en empresas grandes terminan a tiempo, y cumpliendo el presupuesto.
- Las tasas de defectos en productos comerciales se estiman entre 10 y 17 por cada 1000 líneas de código.

Más cifras

- Diciembre de 1990: Miller, Fredrickson. 'An empirical study of the reliability of Unix Utilities' (Communications of the ACM, Vol 33, issue 12, pp.32-44).
  - Entre el 25 y el 33% de las utilidades en Unix podían interrumpirse o colgarse proporcionándoles entradas inesperadas.
- 1995: Miller otra vez, ejecutando `Fuzz` en nueve plataformas tipo Unix diferentes:
  - Fallos entre un 15 y un 43%
  - Muchos fallos ya avisados en el 90 seguían allí
  - La menor tasa de fallos: utilidades de la FSF (7%) y a las incluidas junto con Linux (9%) (¿Uh?)

  No consiguieron hacer fallar ningún servidor de red. Tampoco el servidor *X Window*. Muchos clientes de X, sí

Cifras

- 2000: Miller y Forrester. Fuzz con Windows NT.
  - 45 % de los programas se colgaron o se interrumpieron
  - Enviar mensajes aleatorios Win32 a las aplicaciones hacía fallar al 100%
- 2006: Miller, Cooksey y Moore. Fuzz y Mac OS X.
  - 7% de las aplicaciones de línea de órdenes.
  - De las 30 basadas en GUI sólo 8 no se colgaron o se pararon.

`http://pages.cs.wisc.edu/~bart/fuzz/fuzz.html`

Cifras

Año 2010, CanSecWest.

- Acrobat Reader 9.2.0
- Mac OS X PDF viewer (Mac OS X 10.6)
- iPhone 3.1.2 (sin *jailbreak*). Ver pdfs con el navegador MobileSafari
- OO.org PPT
- MS PowerPoint 2008 para Mac 12.2.3
- MS Office PowerPoint 2007 SP2 MSO (12.0.6425.1000)
  - Resultados similares...

Microsoft ya 'Fuzzea'

`http://www.computerworld.com/s/article/9174539/Microsoft_runs_fuzzing_botnet_finds_1_800_Office_bugs`

Resultados de robustez – 31 dispositivos Bluetooth (2007)

Table 1: Test results from Bluetooth fuzzing with Codenomicon DEFENSICS

Interface/profile	Number of implementations tested with a fuzzer	Number of implementations that failed in the test	Percentage of failed products
L2CAP	31	26	84%
SDP	31	24	77%
RFCOMM	31	28	90%
A2DP	2	2	100%
AVRCP	3	3	100%
HCRP	1	1	100%
HID	1	1	100%
OPP	15	12	80%
FTP	5	5	100%
IRMC Synch	1	1	100%
BIP	1	1	100%
BPP	1	1	100%
HFP	5	2	40%
HSP	5	2	40%
FAX	2	0	0%
DUN	5	2	40%
SAP	4	4	100%

- Sólo 3 dispositivos sobrevivieron a todos los tests.
- Los demás tuvieron problemas con, al menos, un perfil
- La mayoría simplemente se colgaron
- En algunos casos hubo que reprogramar la memoria flash corrupta

Resultados de robustez para 7 puntos de acceso Wifi

Table 2: Failure rates of various Wi-Fi access points when tested with Codenomicon DEFENSICS

	AP1	AP2	AP3	AP4	AP5	AP6	AP7	fail-rate
WLAN	INC	FAIL	INC	FAIL	N/A	INC	INC	33%
IPv4	FAIL	PASS	FAIL	PASS	N/A	FAIL	INC	50%
ARP	PASS	PASS	PASS	N/A	FAIL	PASS	PASS	16%
TCP	N/A	N/A	FAIL	N/A	FAIL	PASS	N/A	66%
HTTP	N/A	PASS	FAIL	PASS	INC	FAIL	FAIL	50%
DHCP	FAIL	FAIL	INC	N/A	FAIL	FAIL	N/A	80%
fail-rate	50%	40%	50%	33%	75%	50%	25%	

Resultados de robustez para 7 puntos de acceso Wifi:

- Sólo se marcan como FAIL los que son reproducibles (INC muestra que ha habido fallos pero no fáciles de repetir).
- Todos fallaron en alguna de las pruebas.

'Wireless Security: Past, Present and Future. Sami Petäjäsoja, Tommi Mäkilä, Mikko Varpiola, Miikka Saukko and Ari Takanen'. Feb 2008.

`http://www.codenomicon.com/resources/whitepapers/Codenomicon_Wireless_WP_v1_0.pdf`

Cifras

- 2004-2005. Honeypot, con varios sistemas (6: Windows, Mac, Linux). Una semana. Fueron escaneados 46255 veces desde el exterior con un resultado de 4892 ataques directos.
  - Windows XP. SP1.
    - 4857 ataques. Comprometido en 18 minutos por Blaster y Sasser. En una hora el ordenador estaba lanzando sus propios ataques.
  - Windows XP. SP2.
    - 16 ataques
    - Sobrevivió a todos ellos
  - MacOS X Jaguar (3, 0), Suse Professional 9.2 (8,0), Fedora Core 3 (8,0), Red Hat 9 (0 ataques).

`http://www.stillsecure.com/docs/StillSecure_DenverPost_Honeypot.pdf`

¿Actualizaciones?

- Feb-Marzo 2005:
  - Menos del 24% de los Windows XP observados en un estudio de AssetMetrix Research Labs tenían SP2.
  - Menos del 7% del total lo tenían.
    251 empresas norteamericanas (seis meses después de su lanzamiento).

Estudio OpenSSH

- Julio 2002 se descubrió un fallo de desbordamiento de memoria remoto
  - Dos semanas después de la publicación del anuncio del fallo, mas de 2/3 de los servidores observados seguían siendo vulnerables.
- Septiembre 2002. Un gusano explotaba el fallo (Slapper).
  - El 60% de servidores era todavía vulnerable.

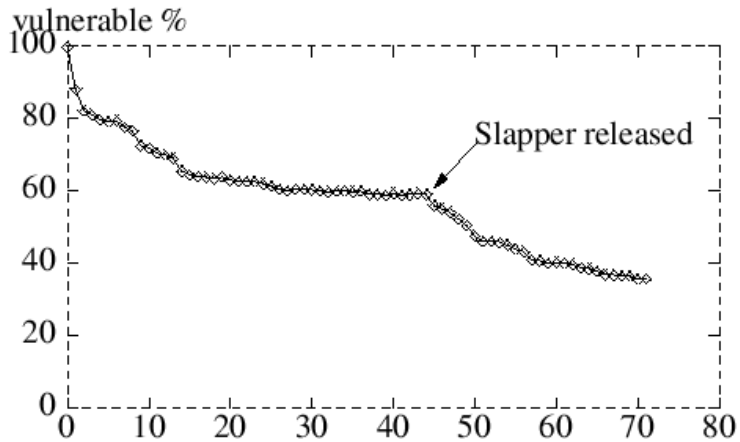


Figure 1 Vulnerable servers over time

'Security holes. . . Who cares? Eric Rescorla'

`http://www.cgisecurity.com/lib/reports/slapper-report.pdf`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

¿Actualizaciones?

Conficker, Downadup

- 23 de octubre de 2008 actualización 'fuera de ciclo'
- 30 días después menos del 50% sin parchear
- 3 meses después 30% sin parchear.

'1 in 3 Windows PCs vulnerable to worm attack'

`http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9126038`

15 de enero de 2009

Introducción. Antes de empezar.

- Los programas no tienen garantía (¿todavía?).
- La seguridad es un problema de gestión de riesgos.
- Pensemos en la seguridad durante el diseño, después ya es tarde.

George Hulme, 'Is It Time For Software Liability?'

`http://www.informationweek.com/security/is-it-time-for-software-liability/229203542`

Geekonomics. The Real Cost of Insecure Software. [David Rice]
Addison-Wesley Professional; 1 edition (December 9, 2007)

Puede haber castigo

Cada vez se habla más de la responsabilidad de las empresas que desarrollan programas:

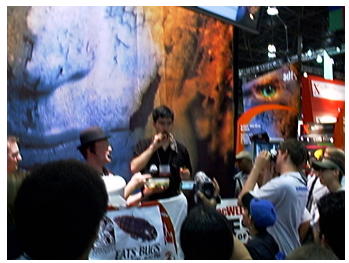
- 1999. Ambrosia Software (Rochester, N.Y.) anunció que si alguno de sus productos requerían la reparación de errores, el responsable de marketing comería insectos en alguna feria.

  `http://www.ambrosiasw.com/ambrosia_times/September_99/EekABug.html`

  > If any of Ambrosia's late 1999 or 2000 releases require bug fixes, Whong may be required to eat live or roasted crickets, grasshoppers, locusts, spiders, or any number of edible insects at a special bug-munching jamboree at MacWorld New York 2000.

  Parece que finalmente tuvieron que comerlos ...

  `http://www.macobserver.com/tmo/article/Ambrosia_President_To_Eat_Live_Bugs_At_MACWORLD/`

- 31 de diciembre de 1999. Las autoridades chinas obligaron a los ejecutivos de la compañía aérea nacional a volar durante esa noche en los vuelos programados.

¿Y los usuarios?

- Cada vez hay más computadores y en más sitios.
- La gente ni sabe ni quiere saber de estos temas.
- Aún peor, saben lo que dicen las noticias.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

¿Y los usuarios?

- Cada vez hay más computadores y en más sitios.
- La gente ni sabe ni quiere saber de estos temas.
- Aún peor, saben lo que dicen las noticias.

Operating System	Malware		Phishing	
	Firefox	Chrome	Firefox	Chrome
Windows	7.1%	23.5%	8.9%	17.9%
MacOS	11.2%	16.6%	12.5%	17.0%
Linux	18.2%	13.9%	34.8%	31.0%

Table 1: User operating system vs. clickthrough rates for malware and phishing warnings. The data comes from stable (i.e., release) versions.

Channel	Malware		Phishing	
	Firefox	Chrome	Firefox	Chrome
Stable	7.2%	23.2%	9.1%	18.0%
Beta	8.7%	22.0%	11.2%	28.1%
Dev	9.4%	28.1%	11.6%	22.0%
Nightly	7.1%	54.8%	25.9%	20.4%

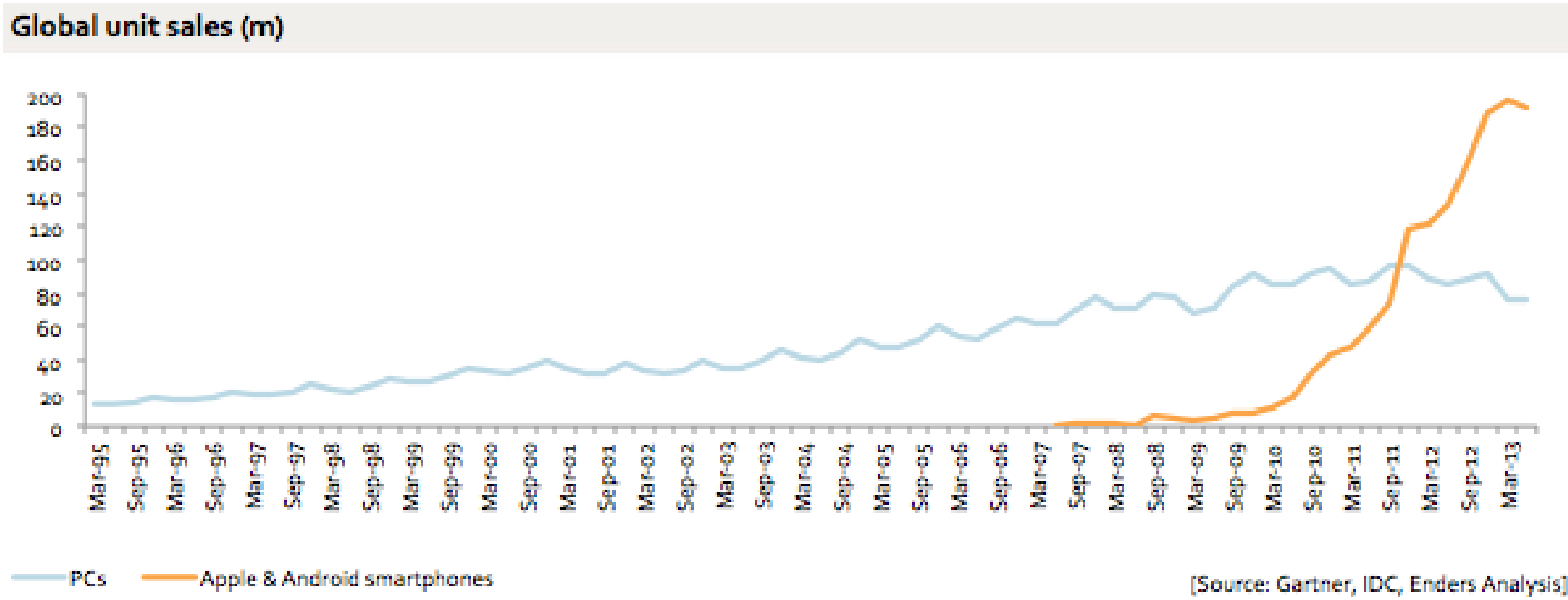
Table 2: Release channel vs. clickthrough rates for malware and phishing warnings, for all operating systems.

D. Akhawe, A. Porter Felt. 'Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness'

`http://www.theregister.co.uk/2013/07/15/google_study_finds_chrome_is_leastsecure_browser/`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

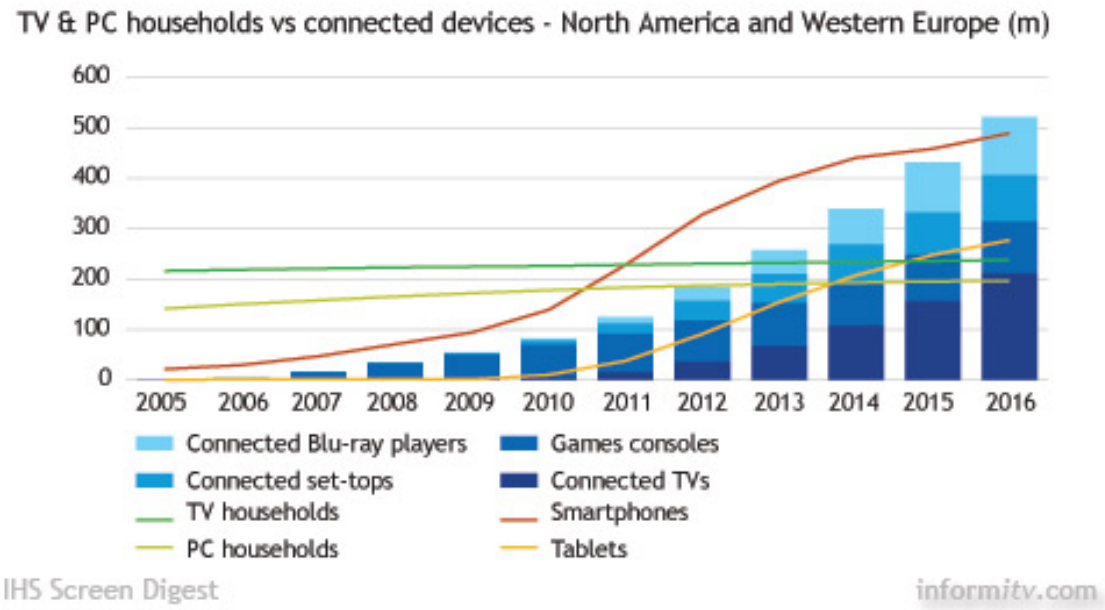
Cada vez más



Benedict Evans. 'PCs and smartphones: the long view'

`http://ben-evans.com/benedictevans/2013/8/23/pcs-and-smartphones-the-long-view`

Son los programas



- Dependemos (mucho) de los computadores (y sus programas).
- El principal problema es que la mayoría de los desarrolladores ni siquiera saben que hay un problema.
- Ni los cortafuegos ni la criptografía resolverán los problemas (el 85% de los avisos del CERT no se pueden prevenir con criptografía).

`http://informitv.com/news/2012/07/15/futureofvideo/`

Future of video advertising in a connected world

Son los programas

- Está bien proteger la transmisión pero los atacantes prefieren los extremos
- Las aplicaciones que interactúan con Internet son las más delicadas, pero no es imprescindible que tengan contacto con la red para ser peligrosas.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Son los programas

- Empezar pronto
- Conocer las amenazas
- Diseñar pensando en la seguridad
- Ceñir el diseño a los análisis de riesgos y las pruebas

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Gestión del riesgo

- La seguridad es un compromiso entre muchos factores:
  - Tiempo hasta que se puede vender
  - Coste
  - Flexibilidad
  - Reutilizabilidad
  - Relaciones entre los anteriores
- Hay que establecer las prioridades, a veces la seguridad no es la principal necesidad.

Seguro o Inseguro

- Mucha gente piensa en la seguridad como algo que se tiene o no se tiene.
- Es muy difícil probar que un sistema de complejidad mediana es seguro.
- Frecuentemente, ni siquiera vale la pena.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Seguro o Inseguro

- Es mas realista pensar en términos de gestión de riesgo:
  - ¿Cuánto riesgo?
  - ¿Cuánto cuesta reducirlo?

Recordar: los 'malos' no crean los defectos, simplemente los utilizan.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Fallos en los programas

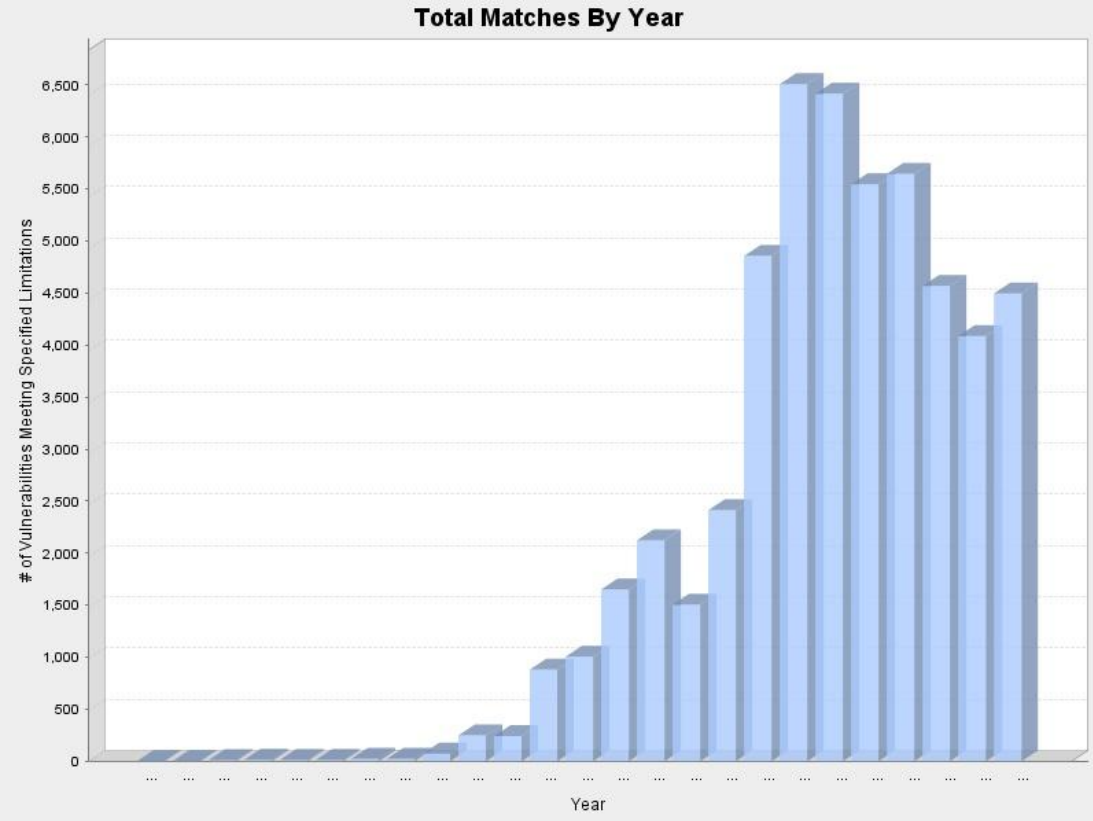
- Año 2000: aproximadamente 20 nuevas vulnerabilidades cada semana
- Muchas en programas con código, pero otras tantas en las que no se conoce
- Unix y Windows también están equilibrados
- Siguen apareciendo problemas en programas probados y usados.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Algunas cifras

NIST: National Institute of Standards and Technology

NVD: National Vulnerabilities Database



Year	Num. of Vulns
1988	2
1989	3
1990	11
1991	15
1992	13
1993	13
1994	25
1995	25
1996	75
1997	252
1998	246
1999	894
2000	1020
2001	1677
2002	2156
2003	1527
2004	2451
2005	4933
2006	6608
2007	6514
2008	(4673) 5632
2009	5733
2010	(4091) 4639
2011	(3451) 4150
2012	(4565) –

28 de octubre de 2012

`http://nvd.nist.gov/statistics.cfm?results=1`

Y más . . . la web

Mike Andrews. 'The State of Web Security'. IEEE Security & Privacy

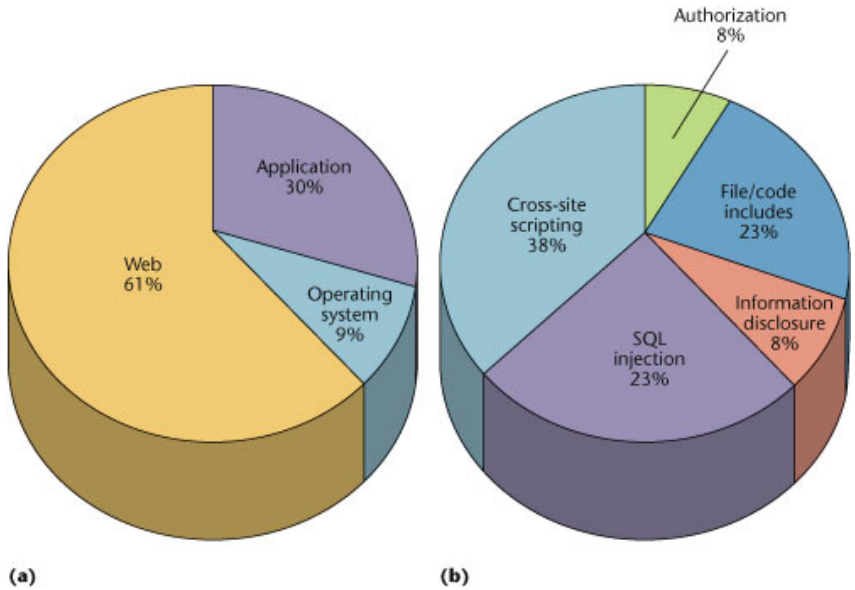


Figure 1. (a) Breakdown of disclosed vulnerabilities by software type in May 2006, and (b) current vulnerability types disclosed in Web-based applications. (Source: SecurityFocus.com)

`http://ieeexplore.ieee.org/iel5/8013/34919/01667997.pdf?arnumber=1667997`

Evolución

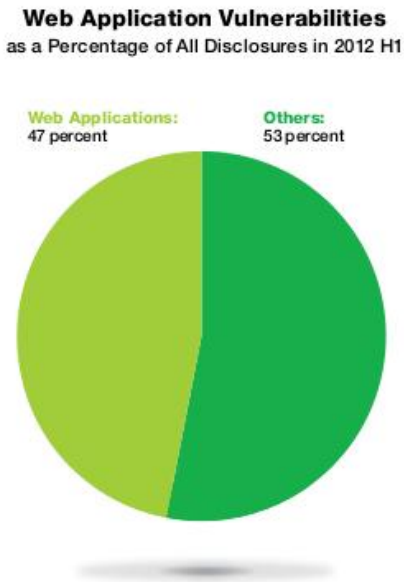


Figure 35: Web Application Vulnerabilities as a Percentage of All Disclosures in 2012 H1

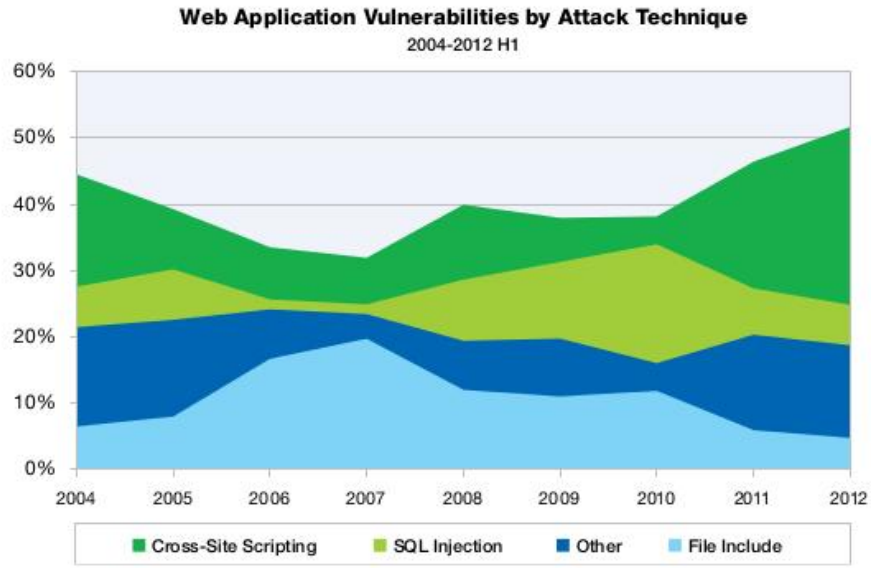
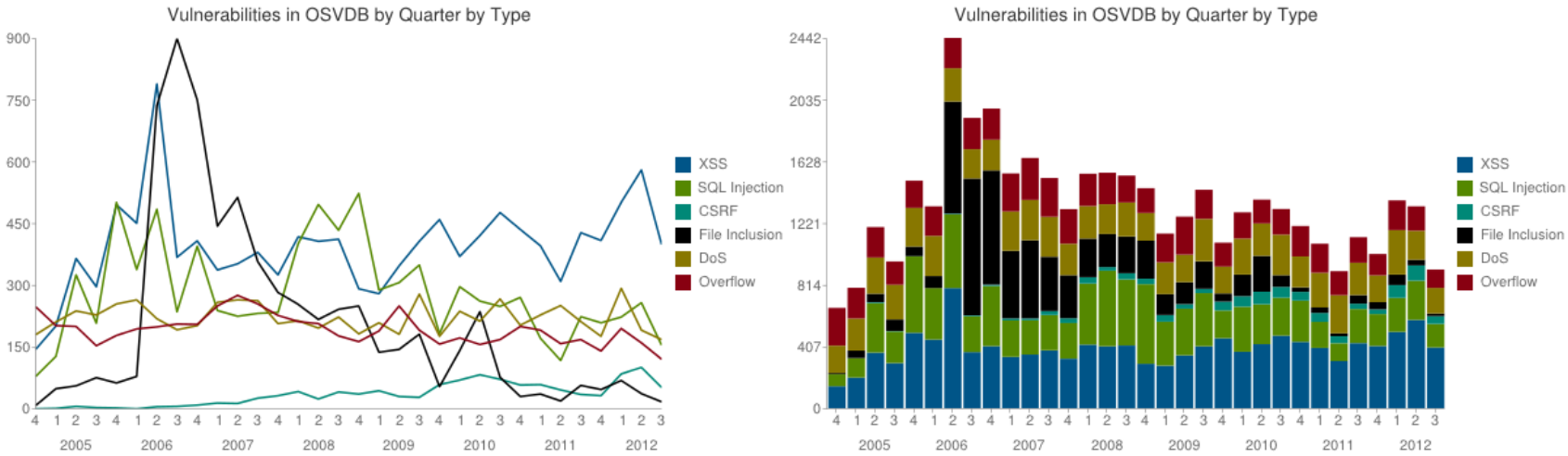


Figure 36: Web Application Vulnerabilities by Attack Technique - 2004-2012 H1

IBM X-Force® 2012 Mid-year Trend and Risk Report

`http://www-935.ibm.com/services/us/iss/xforce/trendreports/`

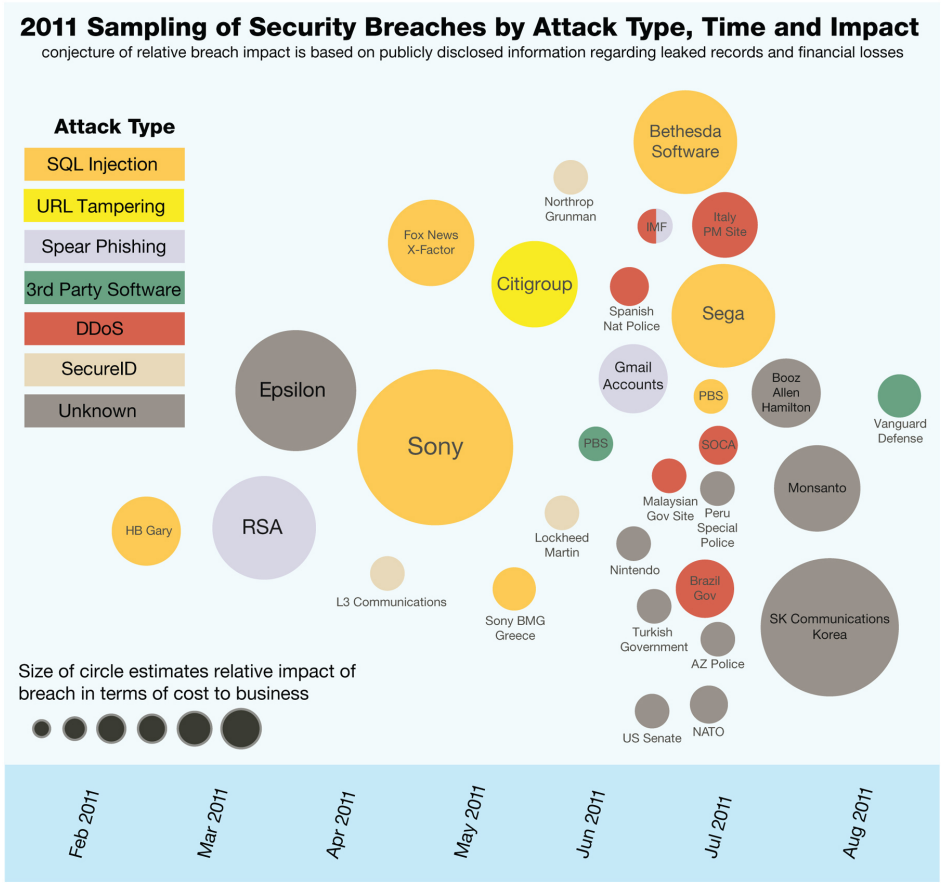
Con más vulnerabilidades



http://osvdb.org/

¿A quién afecta? ¿Qué pasó?

Data Breaches by Attack Type Time Impact.jpg

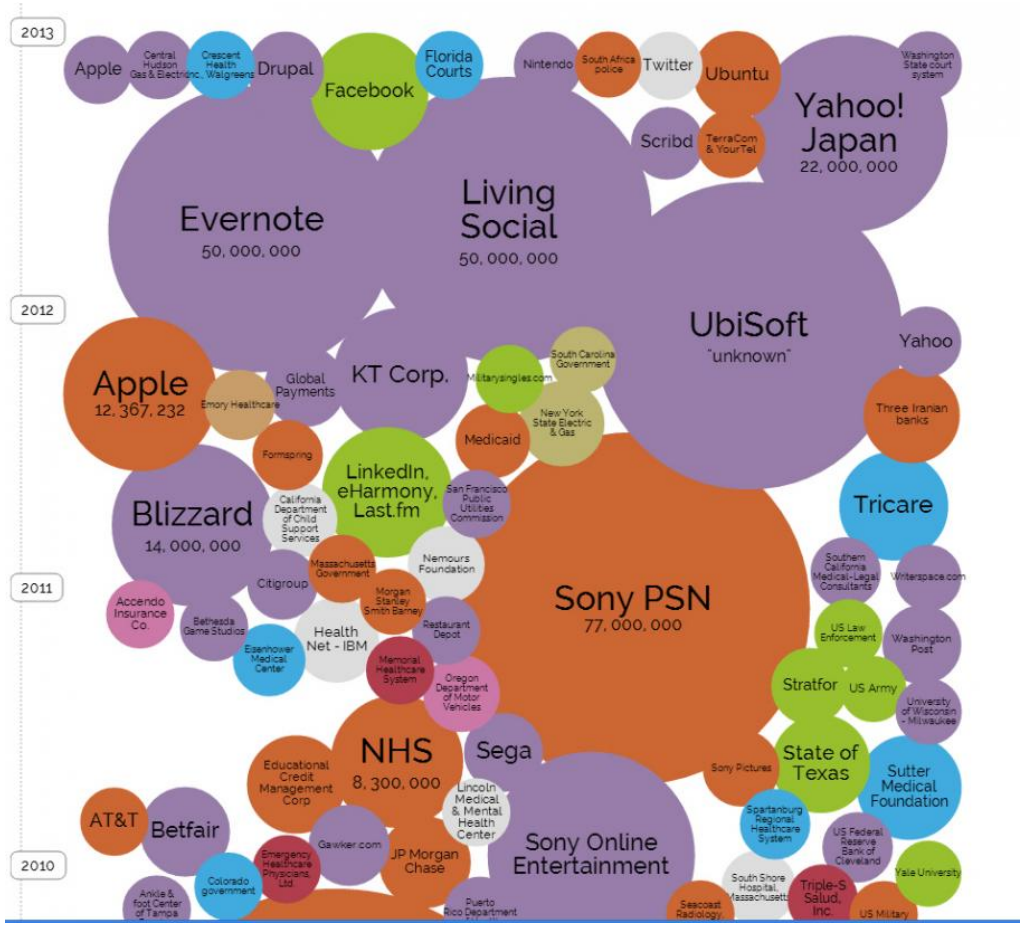


Source: IBM X-Force® Research and Development

IBM X-Force® 2011 Mid-year Trend and Risk Report

http://www-935.ibm.com/services/us/iss/xforce/trendreports/

Robo de datos



Information is Beautiful

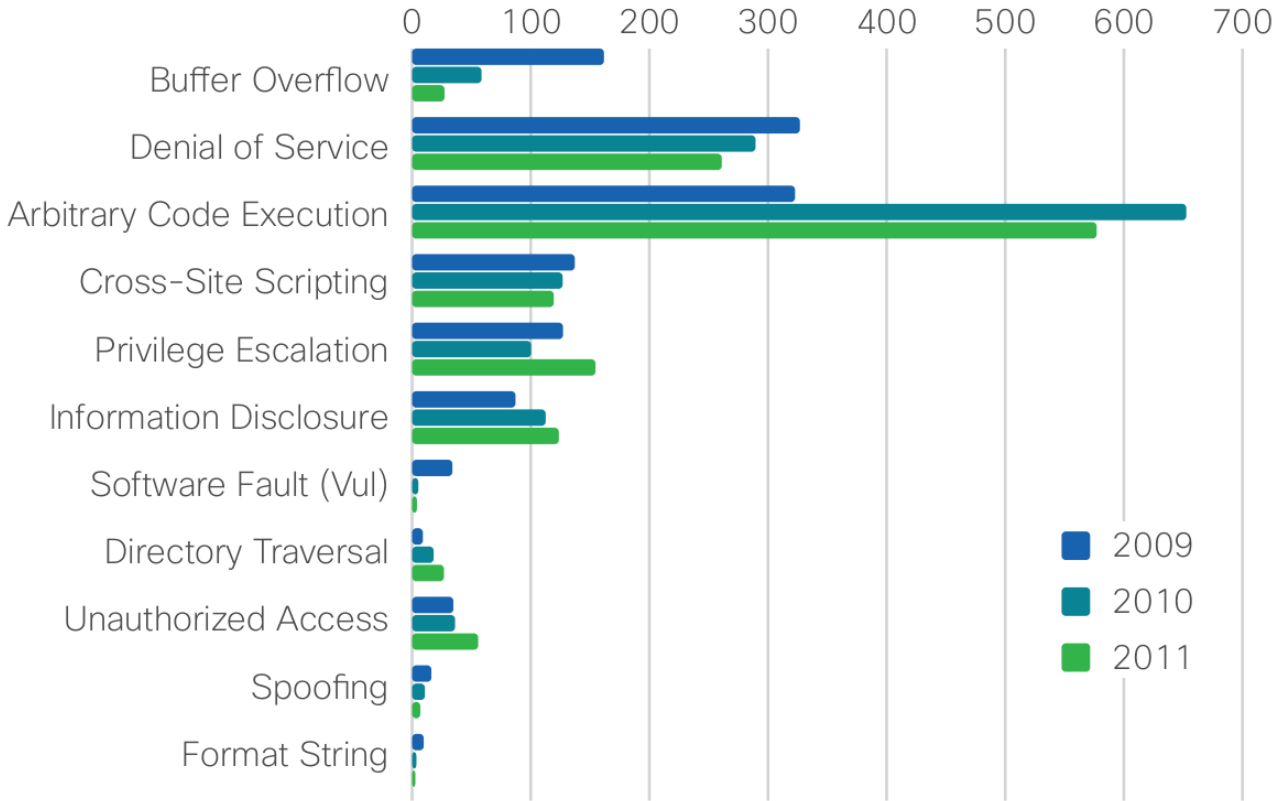
`http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/`

'The Biggest Data Thefts In Recent History [Infographic]'

`http://www.popsci.com/technology/article/2013-07/infographic-biggest-thefts-data-visualized`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Más cifras



`http://cisco.com/en/US/prod/vpndevc/annual_security_report.html`

Vendedores

Ranking	Vendor	Disclosures
1.	Apple	3.8%
2.	Sun	3.3%
3.	Microsoft	3.2%
4.	IBM	2.7%
5.	Oracle	2.2%
6.	Mozilla	2.0%
7.	Linux	1.7%
8.	Cisco	1.5%
9.	Adobe	1.4%
10.	HP	1.2%

Table 3: Vendors with the Most Vulnerability Disclosures, 2009

2009 IBM X-Force Trend and Risk Report

http://www-935.ibm.com/services/us/iss/xforce/trendreports/

Parches

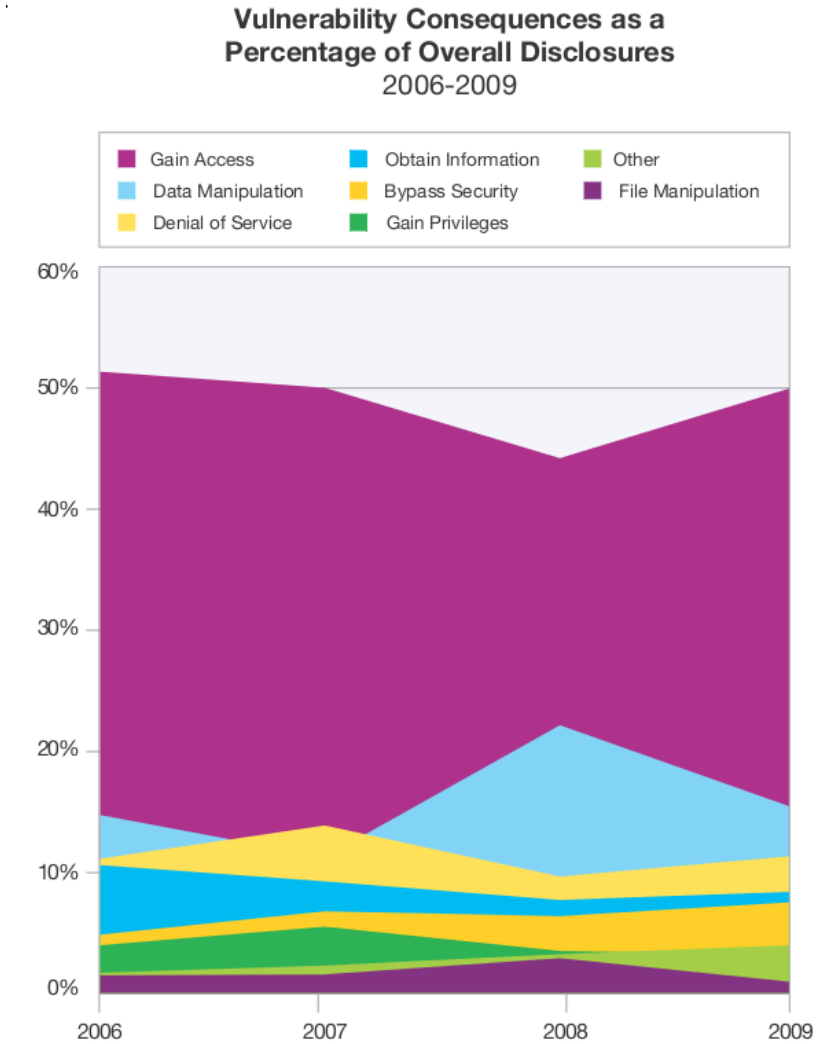
Vendor	Percent of 2009 Disclosures with No Patch	Percent of Critical & High 2009 Disclosures with No Patch
All Vendors–2009 Average	52%	60%
Linux	50%	53%
Oracle	40%	38%
Novell	27%	31%
IBM	25%	27%
Google	47%	25%
Apple	14%	22%
Microsoft	29%	15%
Sun	7%	8%
Symantec	18%	7%
HP	16%	5%
Adobe	4%	4%
Cisco	11%	1%
Opera	47%	0%
GNU	33%	0%
Mozilla	15%	0%
Rim	14%	0%

Table 4: Best and Worst Patchers, 2009

2009 IBM X-Force Trend and Risk Report

http://www-935.ibm.com/services/us/iss/xforce/trendreports/

Consecuencias



2009 IBM X-Force Trend and Risk Report

`http://www-935.ibm.com/services/us/iss/xforce/trendreports/`

¿Dónde conocerlos?

Bases de datos de vulnerabilidades:

- Bugtraq (`http://www.securityfocus.com/`)
- National Vulnerability Database (`http://nvd.nist.gov/`)
- OSVDB, Open Source Vulnerability Database (`http://osvdb.org/`)

Sitios generalistas

- RISKS Digest (`http://catless.ncl.ac.uk/Risks/`)
- Help Net Security (`http://www.net-security.org/`)

CERTs

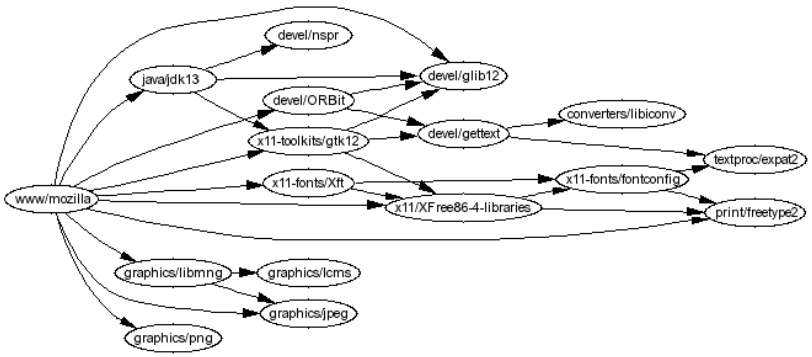
- INTECO, (`http://www.inteco.es/`)
- CERT (Computer Emergency Readiness Team) Advisories (`http://www.cert.org/`)
- IRIS-CERT `http://www.rediris.es/cert/`
- Equipo de Seguridad para la Coordinación de Emergencias en Redes Telemáticas (`http://escert.upc.edu/`)

¿Y las tecnologías?

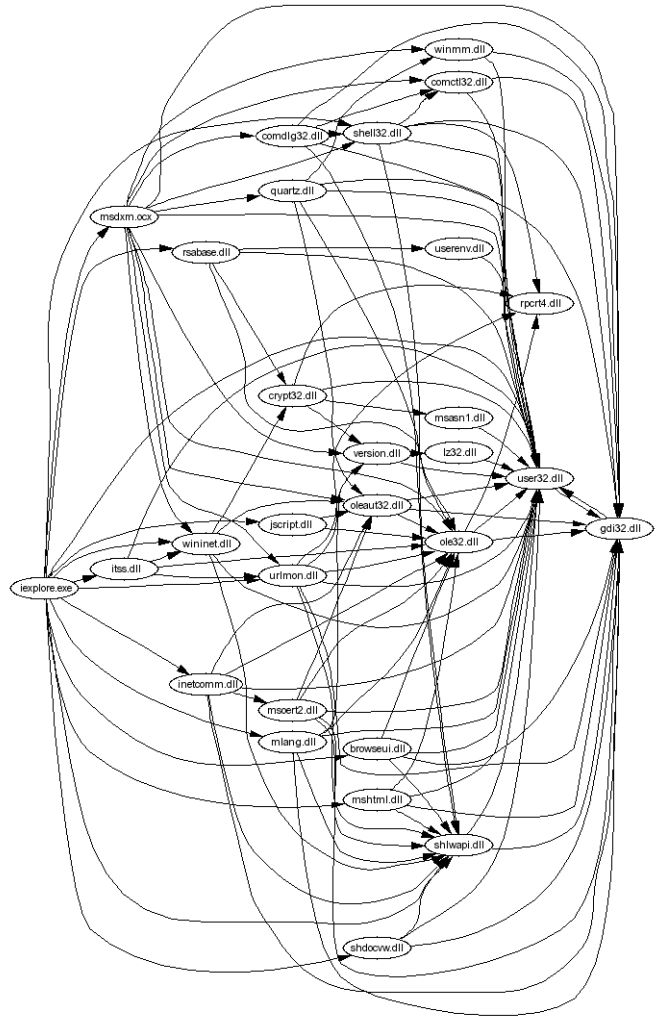
- La complejidad introduce riesgos.
  - Añadir funcionalidades (no presente en el original)
  - Invisibilidad de ciertos problemas
  - Dificultad para analizar, comprender, asegurar.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Complejidad en navegadores



Mozilla 1.3



Explorer 5

http://www.spinellis.gr/blog/20031003/index.html

Complejidad en sistemas

- Windows
  - Windows NT → 35 millones de líneas de código.
  - Windows XP → 40 millones de líneas de código.
  - Windows Vista → 50 millones de líneas de código.
- Unix-Linux
  - Linux 2.2 → 1.78 millones.
    - Linux kernel 3.6 → 15.9 millones.
  - Solaris 7 → 400000.
  - Debian GNU/Linux 2.2 55 millones
    - Debian 5.0 324 millones.
  - Red Hat 6.2 17 millones.
- Mac OS X 10.4 86 millones
  - Mac OS X Darwin 790000 (el kernel)

`http://en.wikipedia.org/wiki/Source_lines_of_code`

Complejidad en sistemas

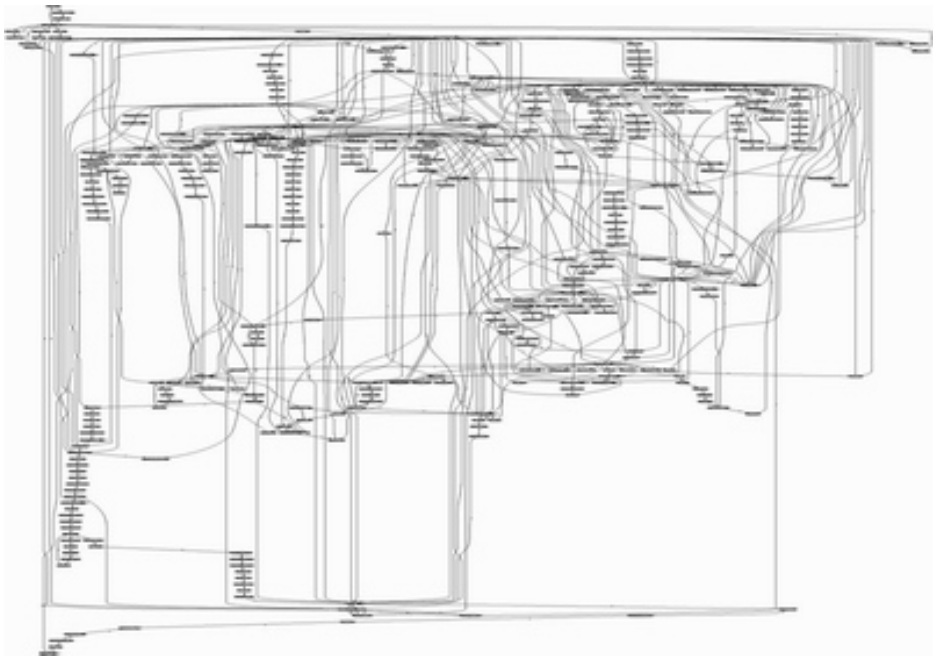
- Windows
  - Windows NT → 35 millones de líneas de código.
  - Windows XP → 40 millones de líneas de código.
  - Windows Vista → 50 millones de líneas de código.
- Unix-Linux
  - Linux 2.2 → 1.78 millones.
    - Linux kernel 3.6 → 15.9 millones.
  - Solaris 7 → 400000.
  - Debian GNU/Linux 2.2 55 millones
    - Debian 5.0 324 millones.
  - Red Hat 6.2 17 millones.
- Mac OS X 10.4 86 millones
  - Mac OS X Darwin 790000 (el kernel)

`http://en.wikipedia.org/wiki/Source_lines_of_code`

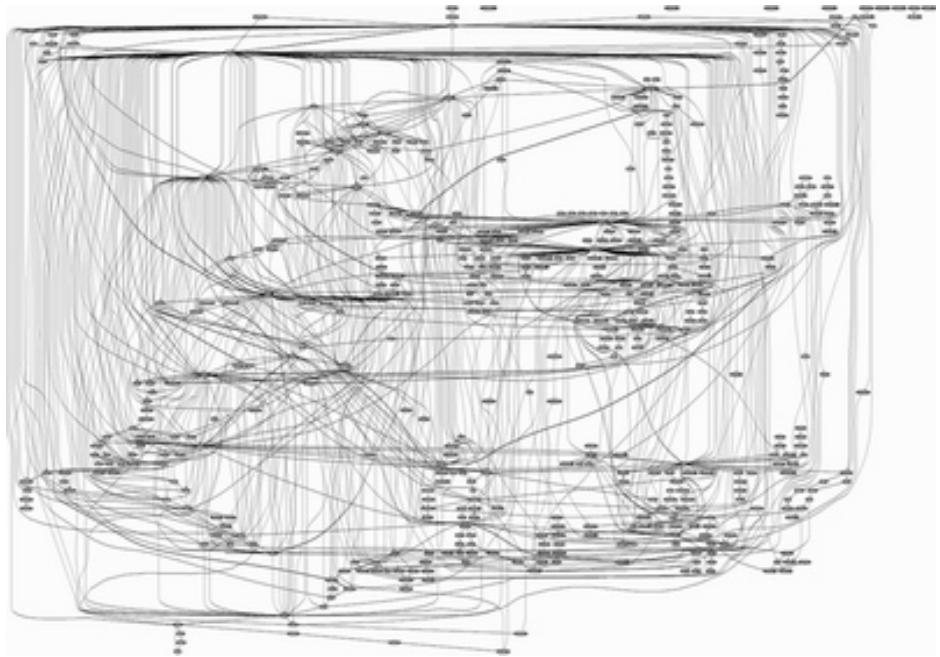
- ¡Seguimos programando en C! (en el mejor de los casos C++)
  - Esto va cambiando . . . Java, .Net, . . .
- Luego hay que instalar, configurar, usar

Complejidad

Linux + Apache



Windows + IIS



http://www.visualcomplexity.com/vc/project.cfm?id=392 http://blogs.zdnet.com/threatchaos/?p=311

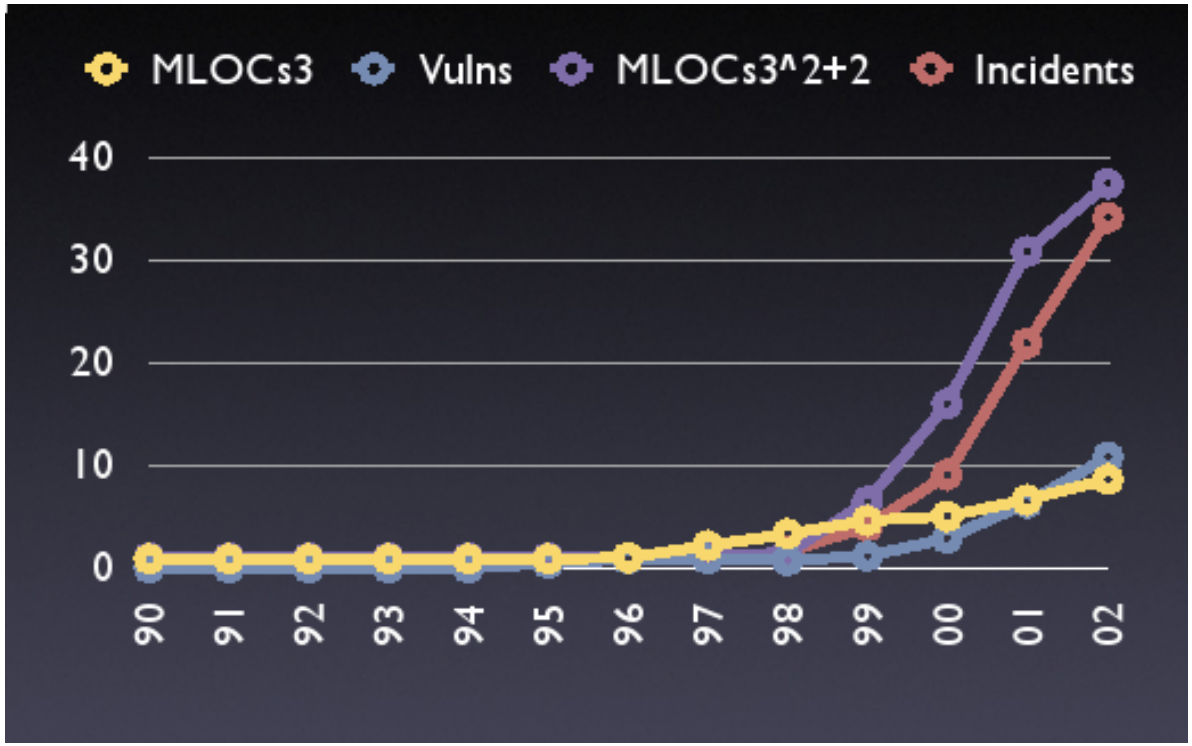
http://web.archive.org/web/20060615055607/http://blogs.zdnet.com/threatchaos/?p=311

'Why Windows is less secure than Linux'

Abril 2006

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Complejidad, vulnerabilidades, incidentes, . . .



MLOCs3 (Three year moving average –media móvil– of code volume)

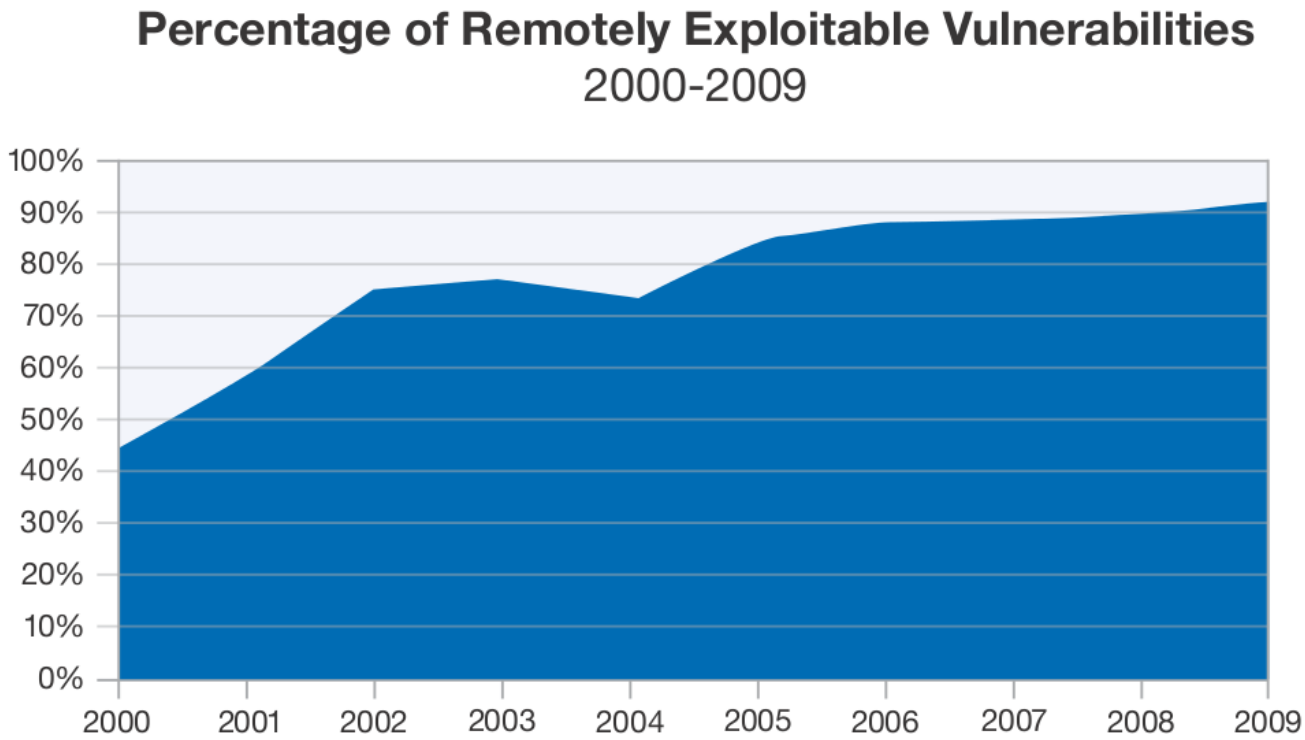
`http://www.stanford.edu/class/msande91si/www-spr04/slides/geer.pdf`

Dan Geer, 2004

'Shared Risk at National Scale. Dan Geer

En red

- Cada vez más redes
  - Los ataques pueden venir de más sitios
  - Ataques automatizados/automáticos
  - Más sitios para atacar, más ataques, mas riesgo



`http://www-935.ibm.com/services/us/iss/xforce/trendreports/`

Extensibilidad

- Código móvil
  - 'Enchufables' en el navegador ('plugins')
  - Módulos, 'drivers'
  - Muchas aplicaciones tienen lenguajes que permiten extenderlas.

Económicamente conveniente (reutilización) pero ...

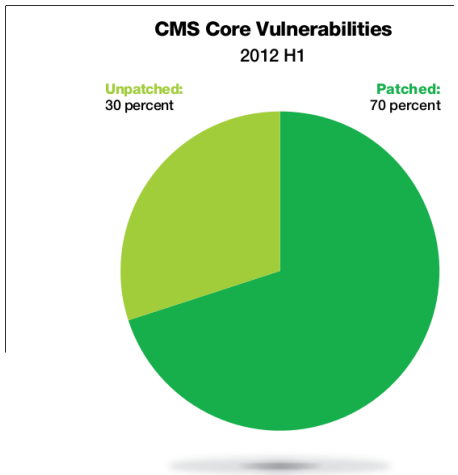


Figure 38: Disclosed Vulnerabilities in core content management systems - unpatched vs. patched - 2012 H1

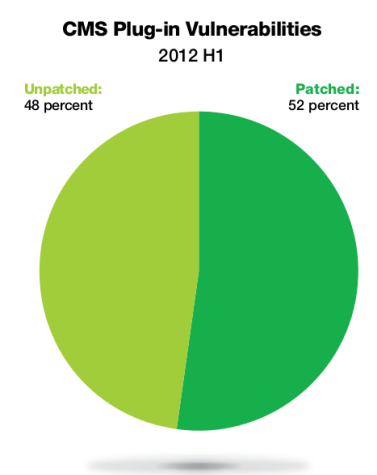


Figure 39: Disclosed vulnerabilities in plug-in content management systems - unpatched vs. patched - 2012 H1

IBM X-Force® 2011 Mid-year Trend and Risk Report

El entorno

- Añadir seguridad a un sistema ya existente es casi imposible
- Es mejor diseñar con la seguridad en mente
- Otra fuente de problemas es 'ambiental': un sistema completamente seguro en el entorno para el que fue diseñado, deja de serlo en otros.

Pero ... ¿Qué es seguridad?

Primero, es importante establecer una política que describa la forma de acceder a los recursos.

- Si no queremos accesos sin autentificar y alguien accede ...
- Si alguien hace un ataque de denegación de servicio ...

Pero ... ¿Qué es seguridad?

Primero, es importante establecer una política que describa la forma de acceder a los recursos.

- Si no queremos accesos sin autentificar y alguien accede ...
- Si alguien hace un ataque de denegación de servicio ...

A veces es evidente lo que está mal, y no hay que hilar tan fino, pero ...

- ¿Un escaneo de puertos es un ataque o no?
- ¿Hay que responder? ¿Cómo?

¿Tiene que ver con la confiabilidad?

'Reliability', confiabilidad, ¿no debería proporcionar seguridad?

- La confiabilidad se mide según la robustez de la aplicación respecto a los fallos.
- La definición de fallo es análoga a la definición de política de seguridad.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

¿Tiene que ver con la confiabilidad?

'Reliability', confiabilidad, ¿no debería proporcionar seguridad?

- La confiabilidad se mide según la robustez de la aplicación respecto a los fallos.
- La definición de fallo es análoga a la definición de política de seguridad.

  - Entonces, la seguridad sería una parte de la confiabilidad: si se puede violar alguna parte de la política de seguridad, hay un fallo.
    Sin embargo...
    - Los problemas de robustez no siempre son problemas de seguridad (Lo son más frecuentemente de lo que se piensa, de todos modos)
    - Si diseñamos pensando en su robustez, seguramente también mejoraremos su seguridad

Malas ideas

Se hacen los programas, se espera a que aparezcan problemas, y se resuelven (si se puede).

- Sólo se resuelven problemas conocidos por los desarrolladores
- No se trabaja ni con el tiempo, ni con la tranquilidad que hace falta.
- Los parches habitualmente atacan al síntoma, no al problema
- Los parches hay que aplicarlos ...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Las metas

- La seguridad no es una característica estática
- 100% seguro no existe (o es mentira)
  Mejor ...
  - ¿Qué queremos proteger?
  - ¿Contra quién?
  - ¿Contra qué?

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Prevención

- Normalmente, se presta atención cuando ya es tarde
- El tiempo en la red es distinto (velocidad)
  - Los ataques se propagan muy rápido
  - Incluso se automatizan

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Trazabilidad, auditabilidad

- Los ataques ocurrirán
- Los contables lo saben (dinero)
- Estas medidas ayudan a detectar, comprender y demostrar los ataques
- Es delicado

Trazabilidad, auditabilidad

- Los ataques ocurrirán
- Los contables lo saben (dinero)
- Estas medidas ayudan a detectar, comprender y demostrar los ataques
- Es delicado

$\Longrightarrow$ Vigilancia

- Auditoría en tiempo real
- Se puede hacer a muchos niveles
  búsqueda de 'firmas', patrones ...
  ... pero también aserciones, código a propósito.
- A menudo, con trampas sencillas se puede capturar a un ladrón, o al menos evitar que haga daño.

Privacidad y Confidencialidad

(Privacidad $\longleftrightarrow$ Intimidad)

- Privacidad y confidencialidad son términos muy relacionados
- Las empresas deben proteger los datos de sus clientes, incluso de los anunciantes
- Los gobiernos también
- No siempre comprendemos bien las consecuencias de nuestras acciones
- Los programas deberían asegurar la privacidad ...
  ... pero los programas sólo sirven para hacer el trabajo
- Si es posible ... no almacenar secretos

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Seguridad multinivel

- Hay secretos 'más secretos' que otros
- Ni las empresas ni los gobiernos quieren que se sepan algunos datos
- Además, no todo el mundo tiene que saber lo mismo ...

  ...
- Es complejo

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Anonimato

- Arma de doble filo
- A veces, hay razones sociales para favorecerlo (enfermedades, persecución política, ...)
- Pero también las hay para controlarla (racismo, terrorismo,..)
- Junto con la privacidad, es de los temas más importantes que hay que decidir.
  - Global Identifier de Microsoft sirve para saber qué copia de MS Office originó un documento
  - WGA (Windows Genuine Advantage)
  - las 'supercookies' de Google y Microsoft ...

Anonimato

- Carnivore, Echelon, ... ¿quién nos garantiza que se usan 'adecuadamente'?
- ¿Y las cookies? ¿Realmente son necesarias? ¿Y si nos las roban?
- Hay empresas que las 'coleccionan' (agrupan, analizan, revenden, ...)
- ¿Y si tenemos que hacerlo nosotros? ¿Qué pasa si algo va mal? ¿La comodidad es compatible con la privacidad?

Autentificación

- Saber *quién* para saber *qué* puede hacer
- Hasta no hace mucho bastaba con la presencia física
- Internet!!!
- `http://mibancofavorito.com`
  - ¿Realmente es MiBancoFavorito$^{(TM)}$?
  - ¿Realmente es un banco?
- SSL da tranquilidad pero ... ¿qué garantiza?

Autentificación

- Nadie mira los datos
- De muchas formas:
  `http://mibancofavorito.com/` $\rightarrow$
  `http://mibacofavorito.com/`
  - ¿Quién se fija?
- Si vale dinero, hay que tener cuidado.
- Algunos esquemas suponen anonimato, otros auditoría.
- Algunos esquemas están orientados a sesiones, otros a transacciones.

Integridad

- Seguir teniendo 'lo mismo'
- Precios, cotizaciones, ... ¿y si nos los cambian?
- La información digital es muy fácil de simular

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Conociendo al enemigo

Es bueno conocer los errores frecuentes, sobre todo porque no se suele hablar mucho del tema.

- Errores de programación (buffers, condiciones de carrera, números aleatorios)
  Pero también ...
- La construcción es importante y también como se usa
  - Arquitectura cliente/servidor
  - Ingeniería social
  - Entradas maliciosas

Las amenazas

- Ver lo que va por la red, ponerse en medio
- Modificar lo que va por la red
- Simular lo que debería ir por la red
- Reemplazar el flujo de datos
- Grabar y repetir

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Las metas de un proyecto

- Funcionalidad (resolver el problema)
- Ergonomía -usabilidad- (a veces la seguridad interfiere con la comodidad/conveniencia)
- Eficiencia (a nadie le gusta esperar)
- El mercado (habitualmente en contra de la simplicidad, y de la gestión de riesgos)
- Simplicidad (buena para los proyectos, buena para la seguridad)

Algunas listas de correo

- Secure Coding `http://www.securecoding.org/list/`
- Web Security `http://lists.webappsec.org/mailman/listinfo/websecurity_lists.webappsec.org`
- WEB APPLICATION SECURITY `http://www.securityfocus.com/archive/107`
- Webappsec (de OWASP): `https://lists.owasp.org/mailman/listinfo/webappsec`
- SECPROG (abandonada? 2005) `http://www.securityfocus.com/archive/98`

En español:

- HACK `https://mailman.jcea.es/options/hacking/`
- Owasp-spanish `https://lists.owasp.org/mailman/listinfo/owasp-spanish`

Bibliografía

- John Viega and Gary McGraw. Building Secure Software. Addison-Wesley
- Michael Howard, David C. LeBlanc. Writing Secure Code. Microsoft Press. Second Edition.
- Innocent Code. A security wake-up call for web programmers. Sverre H. Huseby. Wiley.

- Ross Anderson. Security Engineering. Wiley. Second Edition.

- Computer Security. Dieter Gollmann. Wiley.
- Foundations of Security. What Every Programmer Needs to Know. Christoph Kern , Anita Kesavan , Neil Daswani. Apress.

Bibliografía

- Software Security. Gary McGraw. Addison-Wesley Software Security Series.
- Mark G. Graff, Kenneth R. Van Wyk. Secure Coding: Principles and Practices. O'Reilly & Associates
- John Viega, Matt Messier. Secure Programming Cookbook for C and C++. O'Reilly & Associates.
- Gary McGraw, Edward W. Felten. Securing Java: Getting Down to Business with Mobile Code

Bibliografía

El otro lado.

- Greg Hoglund, Gary McGraw. Exploiting Software. How to break code. Addison Wesley.
- Cyrus Peikari, Anton Chuvakin. Security Warrior. O'Reilly.
- Andrews & Whittaker. How to Break Web Software. Addison Wesley.
- Tom Gallagher; Bryan Jeffries; Lawrence Landauer. Hunting Security Bugs. Microsoft Press.

Más bibliografía

En la red:

- OWASP Guide to Building Secure Web Applications
  `http://www.owasp.org/index.php/OWASP_Guide_Project`
- Security Developer Center Microsoft
  `http://msdn.microsoft.com/security`
  - 'Improving Web Application Security: Threats and Countermeasures'. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan Microsoft Corporation
    (`http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/ThreatCounter.asp`)
- Secure Programming for Linux and Unix HOWTO (¡Uno de los primeros!)
  `http://www.dwheeler.com/secure-programs/`

Hay mas...