

Lección 7: INTRODUCCIÓN A LAS TÉCNICAS DE INYECCIÓN SQL



intypedia
INFORMATION SECURITY ENCYCLOPEDIA

Chema Alonso
chema@informatica64.com

Informática 64
Microsoft MVP en Enterprise Security

Incidentes de Seguridad I: Kaspersky



The screenshot shows the ChannelWeb website interface. At the top, there is a navigation bar with links for Home, Communities, and News. Below this is a secondary navigation bar with categories: News, Reviews, Research, Tools, The IT Channel, Networking, Security, and Storage. The main content area features a red banner for 'NEW ON CHANNELWEB' with a list of articles. The featured article is 'Kaspersky Web Site Hacked With SQL Injection' by Stefanie Hoffman, dated February 9, 2009. The article text describes a security vulnerability in Kaspersky Lab's U.S. website, where a hacker named Unu posted screenshots and a list of tables after a successful SQL injection attack. A quote from the hacker is also included.

ChannelWeb Home | Communities | News
You are not logged into Cha

News | Reviews | Research | Tools | The IT Channel | Networking | Security | Storage

NEW ON CHANNELWEB

- Tech Innovator Winning Products
- Top 100 Executives
- Sign Up For Partner Programs 2010
- Sign Up For Distribution Chiefs 2010
- Sign Up For Channel Chiefs 2010
- FUDWatch Blog
- Coming Up: Women of the Channel Winter Workshop
- Annual Report Cards
- Tech Books Online
- Subscribe to CRN

Kaspersky Web Site Hacked With SQL Injection

By [Stefanie Hoffman](#), ChannelWeb
7:51 PM EST lun. feb. 09, 2009

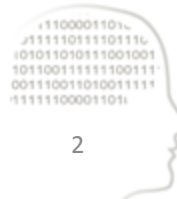
[Discuss This](#)

A security [vulnerability](#) in Moscow-based Kaspersky Lab's U.S. Web site was made public after a [hacker](#) launched a [SQL](#) attack and posted listings of tables contained on the security company's site.

The hacker, known as Unu, posted screen shots as well as a list of tables Feb. 7 to a blog after [hacking](#) into the security company's Web site via a simple SQL injection attack that allowed information to be exposed by entering secret username and password information.

"Kaspersky is one of the leading companies in the security and [antivirus](#) market. It seems as though they are not able to secure their own databases," the hacker said on a [hackerblog.org](#) posting. "Alter one of the parameters and you have access to EVERYTHING: users, activation codes, lists of bugs, admins, shop, etc."

FEATURED VIDEO



Incidentes de Seguridad II: NASA

SC Magazine Virtual Symposium: Botnets

[Home](#) > [News](#) > [NASA sites hacked via SQL injection](#)

NASA sites hacked via SQL injection

[Angela Moscaritolo](#) December 07, 2009



PRINT



EMAIL



REPRINT



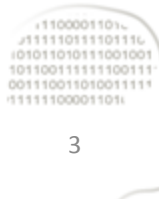
PERMISSIONS

FONT SIZE: [A](#) | [A](#) | [A](#)

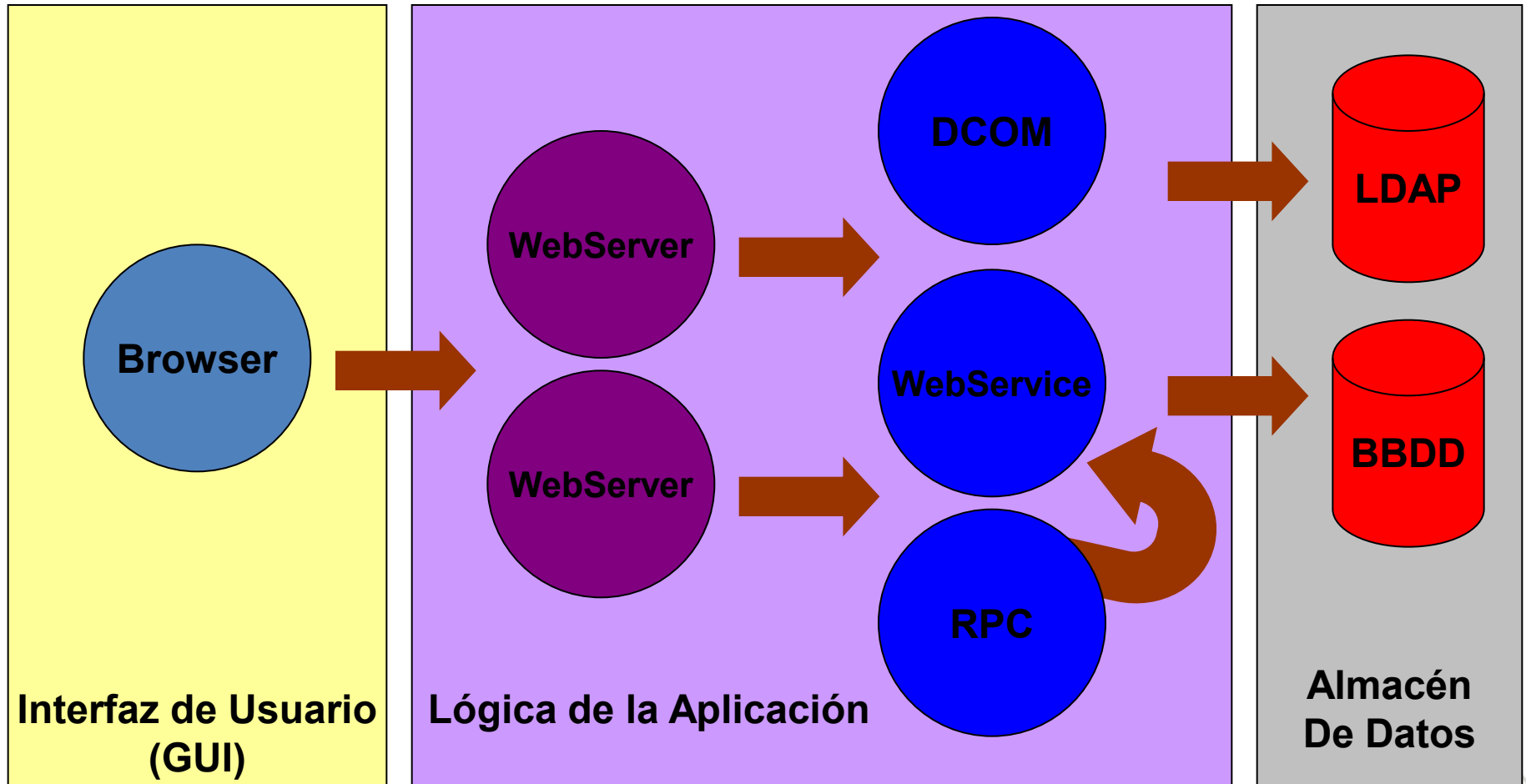
Two NASA sites recently were hacked by an individual wanting to demonstrate that the sites are susceptible to [SQL injection](#).

The websites for NASA's Instrument Systems and Technology Division and Software Engineering Division were accessed by a researcher, who [posted](#) to his blog screen shots taken during the hack.

The researcher, using the alias "c0de.breaker," used [SQL injection](#) to hijack the sites, Gunter Ollmann, VP of research at security firm Damballa, who recently [wrote](#) about the hack, told SCMagazineUS.com on Monday.



Arquitectura de una aplicación Web



OWASP Top 10

OWASP Top 10 – 2007 (Previo)	OWASP Top 10 – 2010 (Nuevo)
A2 – Fallas de inyección	A1 – Inyección
A1 – Secuencia de Comandos en Sitios Cruzados (XSS)	A2 – Secuencia de Comandos en Sitios Cruzados (XSS)
A7 – Pérdida de Autenticación y Gestión de Sesiones	A3 – Pérdida de Autenticación y Gestión de Sesiones
A4 – Referencia Directa Insegura a Objetos	A4 – Referencia Directa Insegura a Objetos
A5 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	A5 – Falsificación de Peticiones en Sitios Cruzados (CSRF)
<T10 2004 A10 – Administración Insegura de Configuración>	A6 – Defectuosa Configuración de Seguridad (NUEVO)
A8 – Almacenamiento Criptográfico Inseguro	A7 – Almacenamiento Criptográfico Inseguro
A10 – Falla de Restricción de Acceso a URL	A8 – Falla de Restricción de Acceso a URL
A9 – Comunicaciones Inseguras	A9 – Protección Insuficiente en la Capa de Transporte
<no disponible en T10 2007>	A10 – Redirecciones y reenvíos no validados (NUEVO)
A3 – Ejecución Maliciosa de Ficheros	<removido del T10 2010>
A6 – Filtrado de Información y Manejo Inapropiado de Errores	<removido del T10 2010>

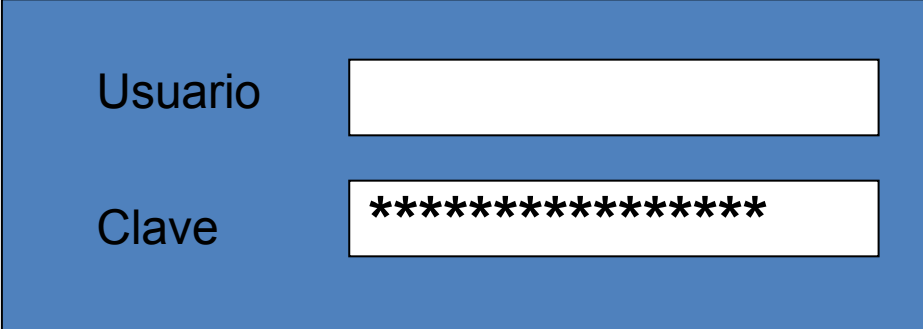
Inyecciones de Código

- Aplicaciones con mala comprobación de datos de entrada.
 - Datos de usuario.
 - Formularios
 - Cookies
 -
 - Datos de llamadas a procedimientos.
 - Links
 - Funciones Scripts
 - Actions
 - ...
- Datos de usuario utilizados en consultas a base de datos.
- Mala construcción de consultas a bases de datos.
- Ataques
 - SQL Injection, LDAP Injection, Xpath Injection
 -



Inyecciones de Código: Ejemplo

- Autenticación de usuario contra base de datos.



A blue rectangular box containing a user authentication form. It has two input fields: 'Usuario' and 'Clave'. The 'Clave' field contains a series of asterisks representing a password.

Usuario	<input type="text"/>
Clave	<input type="password" value="*****"/>

```
Select idusuario from tabla_usuarios  
Where nombre_usuario='$usuario'  
And clave='$clave';
```



Inyecciones de Código: Ejemplo

Usuario	Administrador
Clave	' or '1'='1'

Select idusuario from tabla_usuarios
Where nombre_usuario='Administrador'
And clave="' or '1'='1';



Impacto

- Permiten al atacante:
 - Saltar restricciones de acceso.
 - Elevación de privilegios.
 - Extracción de información de la Base de Datos
 - Parada de SGBDR.
 - Ejecución de comandos en contexto usuario bd dentro del servidor.



Tipos de Ataques: Inbound

- Acceso a información con procedimientos de listado.

`http://www.miweb.com/prog.asp?parametro1=hola`

`http://www.miweb.com/prog.asp?parametro1=' union select nombre, clave,1,1,1 from tabla_usuarios; otra instrucción; xp_cmdshell("del c:\boot.ini"); shutdown --`

o

`http://www.miweb.com/prog.asp?parametro1=1`

`http://www.miweb.com/prog.asp?parametro1=-1 union select; otra instrucción; --`



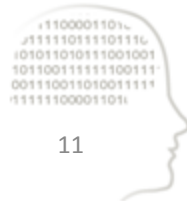
Tipos de Ataques: Outbound

- El atacante vuelca datos utilizando los mensajes de error de la aplicación y el repositorio de datos

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

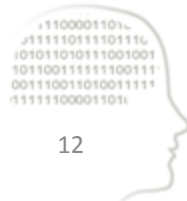
[Microsoft][ODBC SQL Server Driver][SQL Server]Comilla no cerrada antes de la cadena de caracteres 'd'.

/index.asp, line 23



Ataques a ciegas

- La aplicación Web no muestra ningún mensaje de error.
 - No es posible un ataque outbound
- La aplicación no procesa comandos
 - No es posible un ataque inbound
- Se inyectan condicionantes True y False.
Ejemplo:
 - `http://server/miphp.php?id=1 and 1=1`
 - `http://server/miphp.php?id=1 and 1=2`

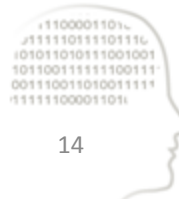


Ataques a ciegas

- ¿Como reconocer diferentes comportamientos?
 - Da un código de error
 - Da una página de error
 - Cambia el hash de la firma
 - Cambia el árbol html
 - Tarda más en responder
 - ...
- Si la página reacciona de forma diferente a inyecciones True y False, entonces se puede extraer datos haciendo búsquedas booleanas
 - `http://www.servidor.com/mostrar_noticias.php?v_id=1 and (100=(select top 1 ascii(substring(login,1,1)) from usuarios))`

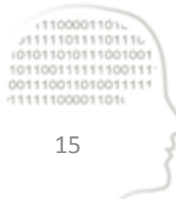
Ataques basados en tiempos

- Si el contenido de la respuesta es idéntico en ambos casos, aún es posible conseguir nuestro objetivo variando condicionalmente el tiempo de respuesta del servidor
 - Si la condición inyectada es verdadera, la aplicación tardará unos segundos en contestar
 - Si la condición es falsa, la aplicación devolverá la misma respuesta, pero en el tiempo habitual
- Se pueden utilizar las mismas técnicas de inyección descritas anteriormente



Inyección SQL basada en tiempos

- ¿Cómo podemos conseguir el retardo?
 - Utilizando instrucciones de retardo implementadas en el propio gestor de bases de datos
 - SQL Server: *waitfor delay*
 - Oracle: *dbms_lock.sleep*
 - MySQL: *sleep*
 - Postgres: *pg_sleep*
 - Utilizando consultas pesadas que consuman muchos recursos del servidor (CPU o memoria)
 - CROSS JOIN que involucren muchas tablas
 - De cualquier otra forma (ej: `xp_cmdshell 'ping...'`)



Contramedidas

- No confianza en medias de protección en cliente.
- Comprobación de datos de entrada.
- Construcción de sentencias SQL con componentes seguros.
- Fortificación de Servidor Web.
 - Códigos de error.
 - Restricción de verbos, longitudes, etc...
 - Filtrado de contenido HTTP en Firewall (WAF).
- Fortificación de SGBD.
 - Restricción de privilegios de motor/usuario de acceso desde web.
 - Aislamiento de bases de datos.





intypedia

INFORMATION SECURITY ENCYCLOPEDIA