HW防守之日志分析 二

LemonSec 今天

转载自：https://www.freebuf.com/column/206352.html
笔者都是作为CTF解题思路来讲述的日志分析方式，其实在真实的网络攻击中，日志分析方式大同小异，这里引荐笔者的文章。

上期为大家介绍了攻击取证之日志分析（一）中的Web日志分析，因此本期将给大家带来系统的日志分析。众所周知，操作系统有很多，但是市面上一般比较主流的操作系统有Windows、Linux以及Mac。其中比较常见的还是Windows以及Linux，Mac毕竟价格有些高昂。在比赛中，系统日志分析的题目更是少之又少，但有时也会结合在一些其他的题目中，因此了解一下也是必要的。接下来，斗哥将从Linux和Windows的系统日志进行讲解。

Linux操作系统

Linux的系统日志一般存放在/var/log目录下，常见的日志（列举部分）有以下：

日志文件	基本详情
/var/log/messages	关于Linux操作系统信息，还包括了系统启动情况等
/var/log/boot.log	系统启动日志
/var/log/lastlog	记录所有用户的近期信息，也可用lastlog命令查看具体内容
/var/log/maillog	邮件日志信息
/var/log/cron	Cron计划任务相关信息的日志
/var/log/secure	系统安全、验证以及授权信息的日志
/var/log/faillog	用户登陆失败信息，包括失败次数、错误登陆命令等
/var/log/btmp	所有登陆失败信息，包括（远程服务、IP地址等）

- **/var/log/messages**

用于记录系统相关信息，如执行程序、系统错误、启动信息等，一般我们会使用message进行查看可疑程序执行的可疑操作，系统在执行程序时出现错误等，具体日志信息如下：

对应的格式：
日期 时间 主机 执行的程序[进程ID]：具体信息

- **/var/log/boot.log**

用于记录系统启动信息的日志，一般用于查看在系统启动时所有相关信息，具体如下：

不难发现，该日志记录的是系统启动时的启动信息，比如开启了哪些服务、做了什么操作都能一目了然。

- **/var/log/lastlog**

用于记录了用户近期登陆情况，直接查看lastlog，可能信息不太明显，但是也可以使用lastlog命令进行查看，会比较详细：

```
[root@localhost log]# cat lastlog
~ts/0192.168.153.1[root@localhost log]# lastlog
用户名           端口     来自             最后登陆时间
root             pts/0    192.168.153.1    日 5月 26 23:23:42 +0800 2019
bin                                        **从未登录过**
daemon                                     **从未登录过**
adm                                        **从未登录过**
lp                                         **从未登录过**
sync                                       **从未登录过**
shutdown                                   **从未登录过**
halt                                       **从未登录过**
mail                                       **从未登录过**
operator                                   **从未登录过**
games                                      **从未登录过**
ftp                                        **从未登录过**
nobody                                     **从未登录过**
systemd-network                            **从未登录过**
dbus                                       **从未登录过**
polkitd                                    **从未登录过**
sshd                                       **从未登录过**
postfix                                    **从未登录过**
chrony                                     **从未登录过**
dockerroot                                 **从未登录过**
```


从上图中可以看出，root用户在5月26日23：23：42登陆到终端，IP为192.168.153.1。

- **/var/log/cron**

Linux的计划任务相关信息的日志，我们也会使用它来找寻攻击者可能会写入的一些恶意计划任务，其中可能会带有一些恶意软件等相关信息。

```
[root@localhost ~]# cat /var/log/cron
Jun 12 11:38:01 localhost run-parts(/etc/cron.daily)[13070]: finished logrotate
Jun 12 11:38:01 localhost run-parts(/etc/cron.daily)[13058]: starting man-db.cron
Jun 12 11:38:01 localhost run-parts(/etc/cron.daily)[13081]: finished man-db.cron
Jun 12 11:38:01 localhost anacron[12674]: Job `cron.daily' terminated
Jun 12 11:49:33 localhost crond[13019]: (CRON) INFO (Shutting down)
Jun 12 11:49:33 localhost crond[13176]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 18% if use
d.)
Jun 12 11:49:33 localhost crond[13176]: (CRON) bad minute (/etc/crontab)
Jun 12 11:49:33 localhost crond[13176]: (CRON) INFO (running with inotify support)
Jun 12 11:49:33 localhost crond[13176]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
Jun 12 11:49:36 localhost crontab[13177]: (root) LIST (root)
Jun 12 11:50:01 localhost crond[13176]: (*system*) RELOAD (/etc/crontab)
Jun 12 11:50:01 localhost crond[13176]: (CRON) bad minute (/etc/crontab)
Jun 12 11:53:08 localhost crond[13176]: (CRON) INFO (Shutting down)
Jun 12 11:53:08 localhost crond[13188]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 13% if use
d.)
Jun 12 11:53:09 localhost crond[13188]: (CRON) INFO (running with inotify support)
Jun 12 11:53:09 localhost crond[13188]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
Jun 12 11:53:26 localhost crontab[13221]: (root) LIST (root)
Jun 12 11:53:29 localhost crond[13188]: (CRON) INFO (Shutting down)
Jun 12 11:53:29 localhost crond[13253]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 94% if use
d.)
Jun 12 11:53:29 localhost crond[13253]: (CRON) INFO (running with inotify support)
Jun 12 11:53:29 localhost crond[13253]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
Jun 12 11:53:30 localhost crontab[13254]: (root) LIST (root)
Jun 12 11:54:01 localhost CROND[13257]: (root) CMD (echo "flag{Loudou}" >> /tmp/test.txt)
Jun 12 11:54:36 localhost crond[13253]: (CRON) INFO (Shutting down)
Jun 12 11:54:36 localhost crond[13267]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 46% if use
d.)
Jun 12 11:54:36 localhost crond[13267]: (CRON) INFO (running with inotify support)
Jun 12 11:54:36 localhost crond[13267]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
Jun 12 11:54:37 localhost crontab[13268]: (root) LIST (root)
```

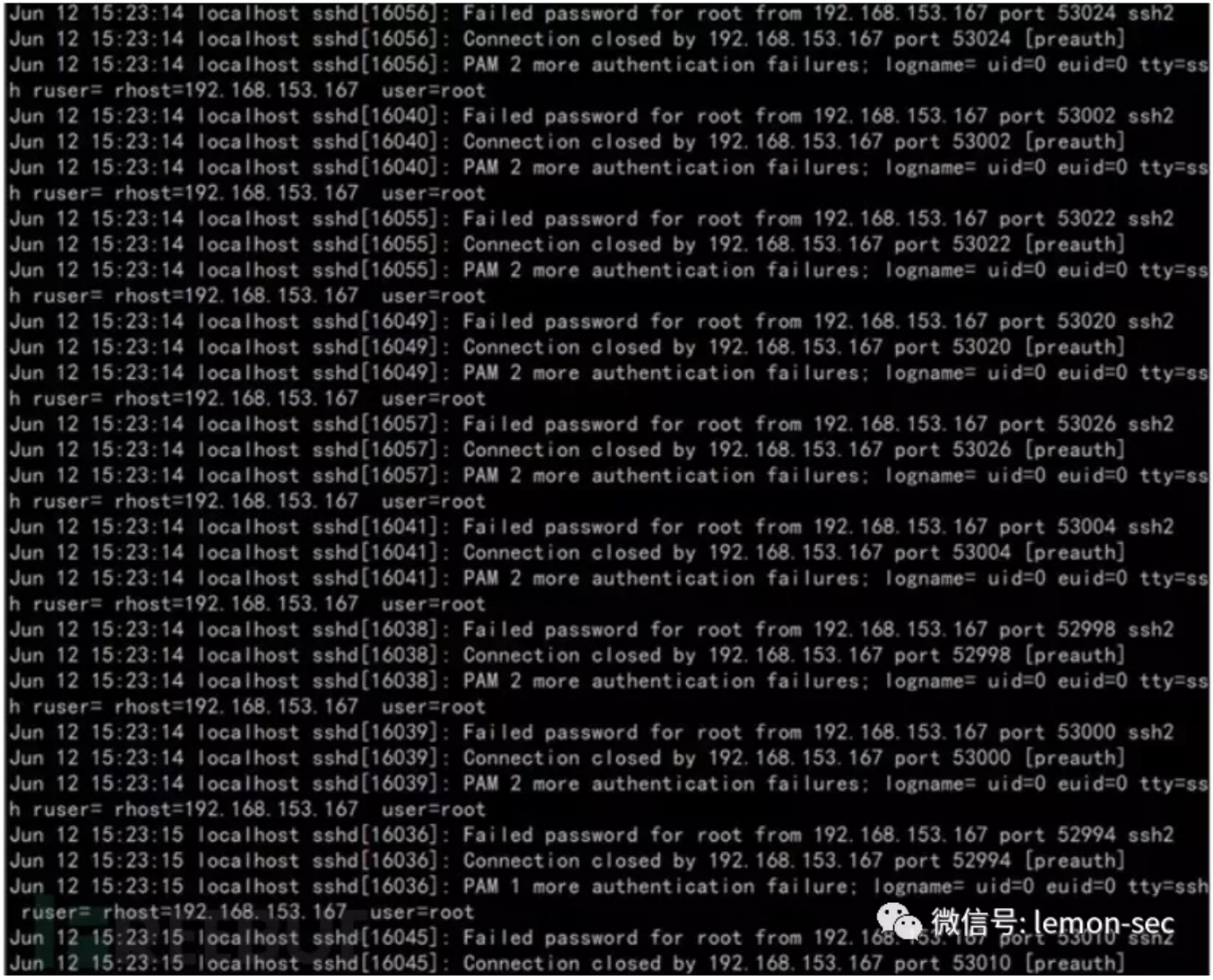

斗哥特意在计划中添加了一个flag，通过cron日志我们可以很明显的看到，有个flag，当然在真实环境或者是CTF比赛中，当然不会这么简单，但是基本上我们排查问题思路也是如此。

- **/var/log/secure**

此日志是linux 的安全日志，被用于记录用户工作的安全相关问题以及登陆认证情况，如：

```
May  5 14:15:09 localhost polkitd[6133]: Registered Authentication Agent for unix-process:16835:902828
(system bus name :1.25 [/usr/bin/pkttyagent --notify-fd 5 --fallback], object path /org/freedesktop/Pol
icyKit1/AuthenticationAgent, locale en_US.UTF-8)
May 14 00:51:49 localhost polkitd[5952]: Loading rules from directory /etc/polkit-1/rules.d
May 14 00:51:49 localhost polkitd[5952]: Loading rules from directory /usr/share/polkit-1/rules.d
May 14 00:51:49 localhost polkitd[5952]: Finished loading, compiling and executing 2 rules
May 14 00:51:49 localhost polkitd[5952]: Acquired the name org.freedesktop.PolicyKit1 on the system bus
May 14 00:51:51 localhost sshd[6692]: Server listening on 0.0.0.0 port 22.
May 14 00:51:51 localhost sshd[6692]: Server listening on :: port 22.
May 14 00:54:16 localhost login: pam_unix(login:session): session opened for user root by LOGIN(uid=0)
May 14 00:54:16 localhost login: ROOT LOGIN ON tty1
May 14 00:56:13 localhost sshd[16918]: Accepted password for root from 192.168.153.1 port 1935 ssh2
May 14 00:56:14 localhost sshd[16918]: pam_unix(sshd:session): session opened for user root by (uid=0)
May 14 01:00:32 localhost unix_chkpwd[16983]: password check failed for user (root)
May 14 01:00:32 localhost sshd[16981]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid
=0 tty=ssh ruser= rhost=192.168.153.1  user=root
May 14 01:00:32 localhost sshd[16981]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
user "root"
May 14 01:00:34 localhost sshd[16981]: Failed password for root from 192.168.153.1 port 2005 ssh2
May 14 01:00:38 localhost unix_chkpwd[16984]: password check failed for user (root)
May 14 01:00:38 localhost sshd[16981]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
user "root"
May 14 01:00:39 localhost sshd[16981]: Failed password for root from 192.168.153.1 port 2005 ssh2
May 14 01:00:42 localhost sshd[16981]: error: Received disconnect from 192.168.153.1 port 2005:13: The
user canceled authentication.  [preauth]
May 14 01:00:42 localhost sshd[16981]: Disconnected from 192.168.153.1 port 2005 [preauth]
May 14 01:00:42 localhost sshd[16981]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh
 ruser= rhost=192.168.153.1  user=root
May 16 21:07:21 localhost polkitd[5952]: Registered Authentication Agent for unix-process:17682:382250
(system bus name :1.88 [/usr/bin/pkttyagent --notify-fd 5 --fallback], object path /org/freedesktop/Pol
icyKit1/AuthenticationAgent, locale en_US.UTF-8)
May 22 17:17:29 localhost polkitd[6101]: Loading rules from directory /etc/polkit-1/rules.d
May 22 17:17:29 localhost polkitd[6101]: Loading rules from directory /usr/share/polkit-1/rules.d
May 22 17:17:29 localhost polkitd[6101]: Finished loading, compiling and executing 2 rules
May 22 17:17:29 localhost polkitd[6101]: Acquired the name org.freedesktop.PolicyKit1 on the system bus
May 22 17:17:31 localhost sshd[6734]: Server listening on 0.0.0.0 port 22.
May 22 17:17:31 localhost sshd[6734]: Server listening on :: port 22.
```


不难发现，上面记录了一些服务如polkitd、login、sshd等，无论成功与否，均会被记录到此日志中，有时我们也可以通过它来判断服务器是否被攻击（如暴力破解、调用一些系统方法等），以下举个被爆破之后的日志：



可以从上图中很容易发现该服务器正在被IP为192.168.153.167的攻击机在短时间内对root用户进行多次尝试ssh登录。

讲完Linux，就得讲一讲Windows了，Windows大家肯定比较熟悉，因为我们现在的笔记本也基本都是Windows操作系统，但是说起查日志，可能还是相对比较少，但在Windows服务器中，日志还是挺关键的，确切的说不管在什么操作系统中，日志都是很重要的。

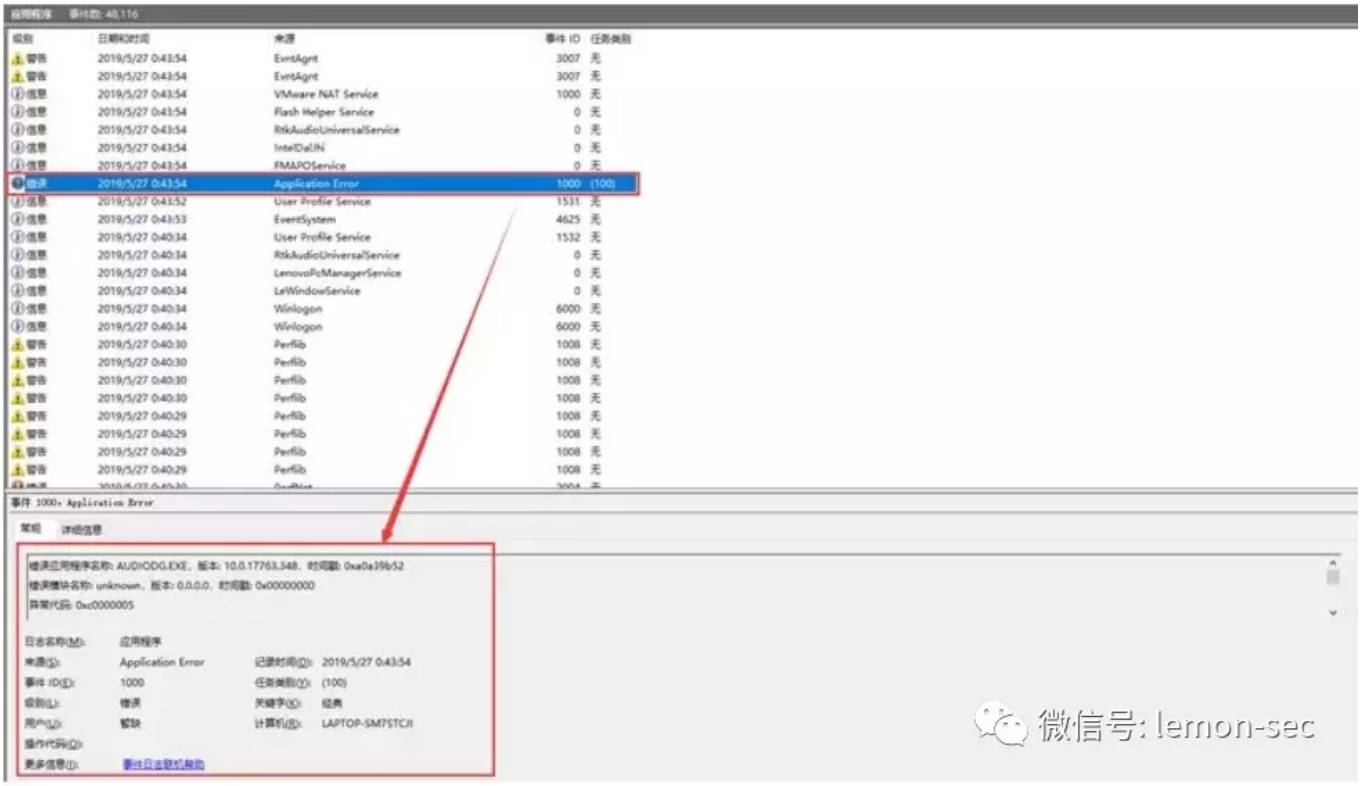
话不多说，开始和大家一起分析分析Windows日志。

Windows操作系统

Windows日志一般在事件查看器中可以进行查看，通常分为五个：应用程序、安全、Setup、系统、转发事件。并且这五个中又以应用程序、安全以及系统日志较为常见，因此在本期中，将介绍这三个。

- **应用程序日志**

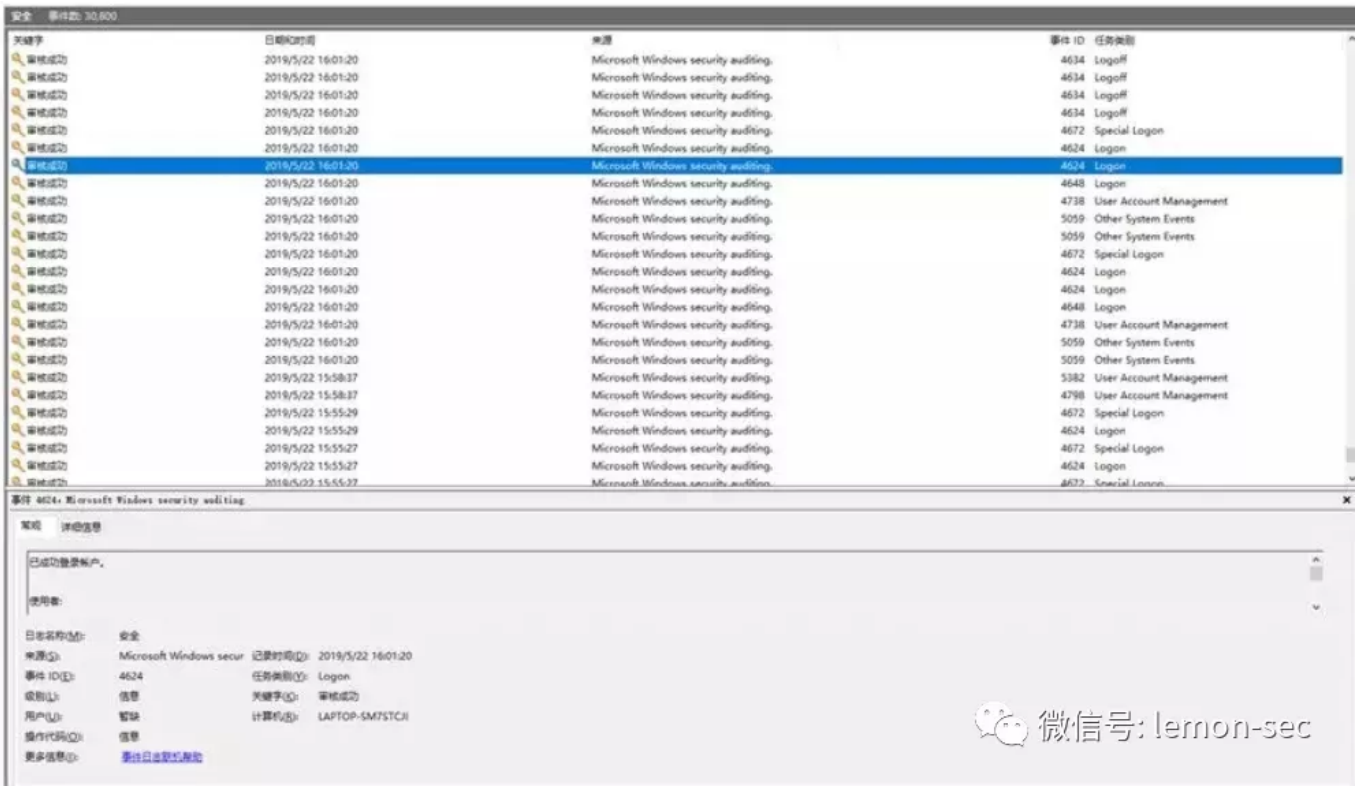
此日志顾名思义便是记录了应用程序的运行情况，包括运行出错、甚至于出错的原因，如：



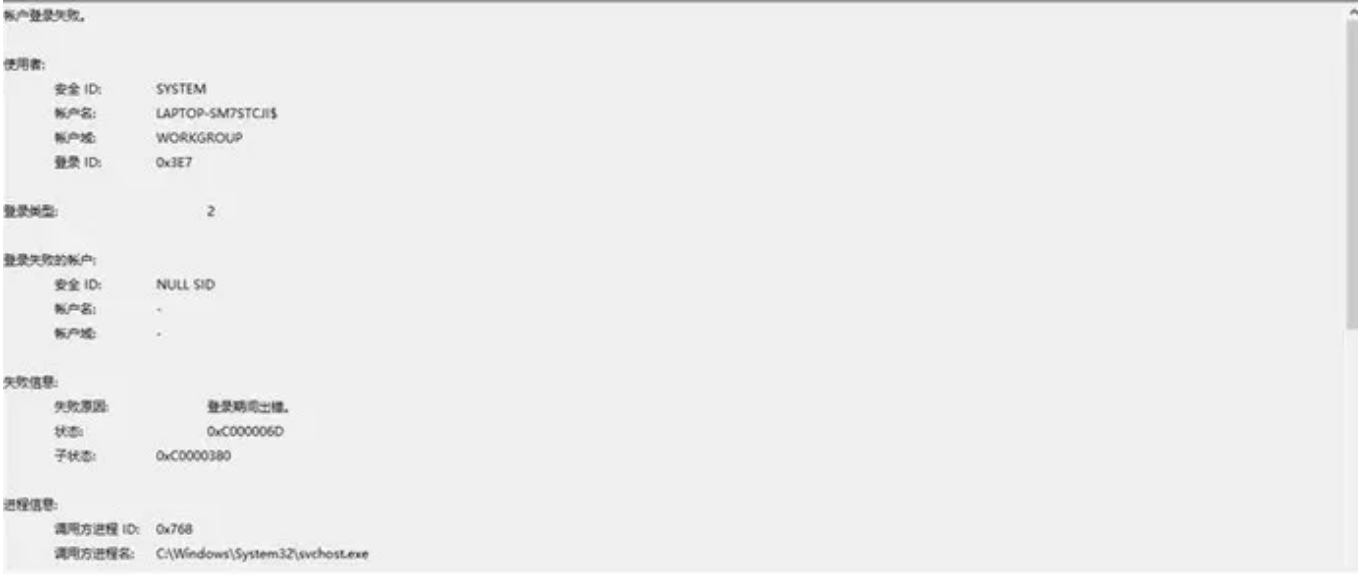
它指出了错误应用程序名称、版本、具体时间错，并且还指出了错误的模块以及异常代码，故而，我们可以通过这些信息，进行对应的故障排查，具体如何排查可通过适当的资料等进行，斗哥在此便不做过多说明，需要提的是它在Windows中保存在Application.evtx文件中，如果在CTF比赛中，看到这个文件，那么可能就是让你进行应用程序日志分析了。

- **安全日志**

此处的安全日志和Linux的安全日志相似，但是它只记录用户登陆情况、用户访问时间以及访问是否授权等，通过它我们可以轻松的发现是否存在爆破风险（一般在短时间内发现大量登陆失败，即可认为该账号被爆破了）。



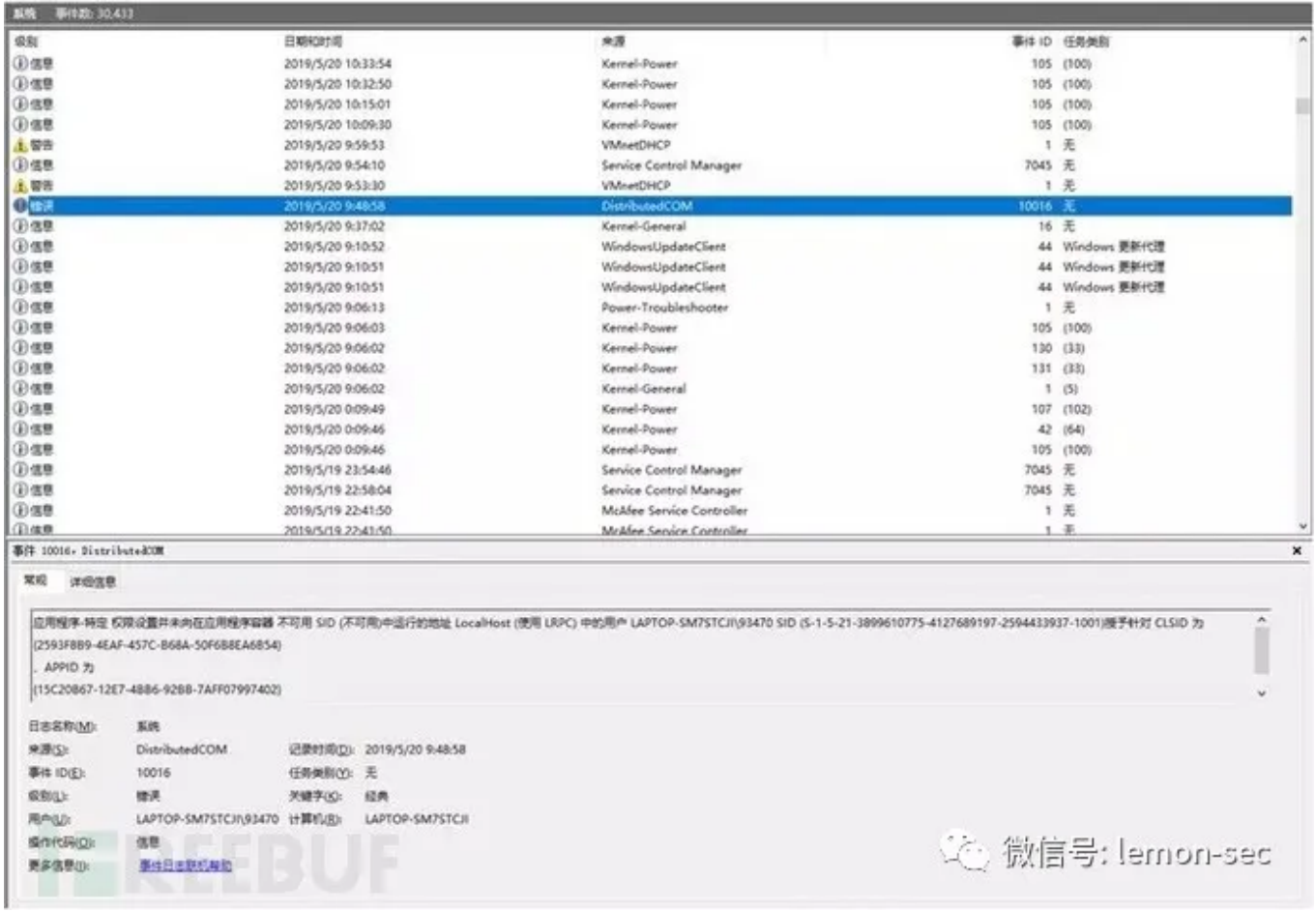
上图显示的是正常的日志，并且它所给的信息也非常详细（以一个登陆失败为例）。



它详细到可以发现使用者信息、登陆类型、登陆失败的账户、失败信息、进程信息、内网信息以及详细身份验证信息等，十分方便。它在操作系统中保存在Security.evtx文件下，我们也可以通过双击它打开安全日志。

- **系统日志**

系统日志则是记录了操作系统安装的应用程序软件相关的事件。它包括了错误、警告及任何应用程序需要报告的信息等。



相比于Linux 的日志，Windows对于系统日志的记录，也是挺详细的，我们可以通过它来进行一些分析判断，它存在于System.evtx文件中。

本期小结

本期的日志分析就介绍到此，主要为系统日志分析，这在分析取证中还是蛮重要的。