实验 - 查找日志文件

1. 简介

在本实验中，同学们将熟悉如何查找和操作 Linux 日志文件。

1. 所需资源

* CyberOps Workstation 虚拟机

1. 日志文件概述

日志文件（又写作 logfile）是计算机用来记录事件的文件。事件可能由软件程序、后台进程、服务或服务之间的事务（包括操作系统本身）生成。日志文件依赖于生成日志文件的应用。日志文件是否遵守相关约定由应用开发人员决定。软件文档中应包含其日志文件的相关信息。

* 1. Web 服务器日志文件示例

由于日志文件本质上是跟踪特定事件的一种方式，因此存储的信息类型因生成事件的应用或服务而异。

* + 1. 请考虑下面的单个日志条目。它由最常用的 Web 服务器 Apache 生成。

[Wed Mar 22 11:23:12.207022 2017] [core:error] [pid 3548:tid 4682351596] [client 209.165.200.230] File does not exist: /var/www/apache/htdocs/favicon.ico

上面的单个日志条目表示 Apache 的 Web 事件记录器。在 Web 事务中有些信息很重要，包括客户端 IP 地址、事务的时间和详细信息。上面的条目可以分解为五个主要部分：

**时间戳：**此部分记录发生事件的时间。服务器时钟必须正确同步，才能准确地交叉引用和追溯事件。

**类型：**表示事件的类型。在本例中，类型为错误。

**PID：**包含 Apache 当前使用的进程 ID 的相关信息。

**客户端：**记录请求客户端的 IP 地址。

**说明：**包含事件的说明。

基于上面的日志条目，描述发生的事件。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

使用下面的 **cat** 命令列出 Web 服务器示例日志文件。示例文件位于 /var/log 中：

[analyst@secOps ~]$ **cat /var/log/logstash-tutorial.log**

83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1" 200 203023 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_9\_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"

83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-dashboard3.png HTTP/1.1" 200 171717 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_9\_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"

83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monitorama-2013/plugin/highlight/highlight.js HTTP/1.1" 200 26185 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_9\_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36”

<some output omitted>

上面的输出是否仍然被认为是一项 Web 事务？解释为什么 **cat** 命令的输出格式与 (a) 项中所示的单个条目不同。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. 操作系统日志文件示例

任何软件都可以保留日志文件，包括操作系统本身。通常，Linux 使用 /var/log 目录来存储各种日志文件，包括操作系统日志。现代操作系统包含一系列复杂的软件，因此使用若干不同的文件来记录事件。此部分简要介绍 /var/log/messages 文件。

* + 1. 消息文件存储在 /var/log 下，用于存储各种系统事件。连接新的 USB 驱动器，网卡变得可用，失败的 root 登录尝试次数过多等，都属于记录到 /var/log/messages 文件中的一些事件。使用 **more** 命令显示 /var/log/message 文件的内容。与 **cat** 命令不同，**more** 用于对文件进行逐步导航。按 **ENTER** 键可逐行推进，按 **SPACE** 键可推进整个页面。按 **q** 或 **CTRL + C** 可中止并退出 **more**。同学们的条目将与下面所示的输出不同。

**注**：由于消息文件属于 root 用户，因此需要使用 **sudo** 命令。

[analyst@secOps ~]$ **sudo more /var/log/messages**

[sudo] password for analyst:

Mar 20 08:34:38 secOps kernel: [ 6.149910] random: crng init done

Mar 20 08:34:40 secOps kernel: [ 8.280667] floppy0: no floppy controllers found

Mar 20 08:34:40 secOps kernel: [ 8.280724] work still pending

Mar 20 08:35:16 secOps kernel: [ 44.414695] hrtimer: interrupt took 5346452 ns

Mar 20 14:28:29 secOps kernel: [21239.566409] pcnet32 0000:00:03.0 enp0s3: link down

Mar 20 14:28:33 secOps kernel: [21243.404646] pcnet32 0000:00:03.0 enp0s3: link up, 100Mbps, full-duplex

Mar 20 14:28:35 secOps kernel: [21245.536961] pcnet32 0000:00:03.0 enp0s3: link down

Mar 20 14:28:43 secOps kernel: [21253.427459] pcnet32 0000:00:03.0 enp0s3: link up, 100Mbps, full-duplex

Mar 20 14:28:53 secOps kernel: [21263.449480] pcnet32 0000:00:03.0 enp0s3: link down

Mar 20 14:28:57 secOps kernel: [21267.500152] pcnet32 0000:00:03.0 enp0s3: link up, 100Mbps, full-duplex

Mar 20 14:29:01 secOps kernel: [21271.551499] pcnet32 0000:00:03.0 enp0s3: link down

Mar 20 14:29:05 secOps kernel: [21275.389707] pcnet32 0000:00:03.0 enp0s3: link up, 100Mbps, full-duplex

Mar 22 06:01:40 secOps kernel: [ 0.000000] Linux version 4.8.12-2-ARCH (builduser@andyrtr) (gcc version 6.2.1 20160830 (GCC) ) #1 SMP PREEMPT Fri Dec 2 20:41:47 CET 2016

Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'

Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'

Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'

Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: xstate\_offset[2]: 576, xstate\_sizes[2]: 256

Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.

Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Using 'eager' FPU context switches.

<some output omitted>

请注意，上列事件与 Web 服务器事件有很大的不同。由于是操作系统本身生成了此日志，因此所有记录的事件都与操作系统本身有关。

* + 1. 如有必要，输入 **Ctrl + C** 退出上个命令。
    2. 日志文件对故障排除非常重要。假设该特定系统的用户报告下午 2:30 左右所有网络运营都很缓慢。同学们能在上面显示的日志条目中找到证据吗？如果能的话，在哪些行中可以找到证据？请说明原因。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 在未知系统中查找日志文件

CyberOps Workstation 虚拟机包括轻型 Web 服务器 nginx。此部分将介绍如何使用 CyberOps Workstation 虚拟机查找和显示 nginx 日志。

**注**：nginx 使用默认设置安装在 CyberOps Workstation 虚拟机上。使用默认设置时，其全局配置文件位于 /etc/nginx/nginx.conf 下，访问日志文件位于 /var/log/nginx/access.log 中，并且错误被重定向到终端窗口。但是，安全分析师通常会使用工具和服务安装详情未知的计算机。此部分介绍查找描述 nginx 的此类文件的过程，但绝不是详尽的操作说明。不过，对于在不熟悉的系统上查找和显示日志文件，这应当是一项很好的练习。

* + 1. 使用新软件时，第一步是查看软件文档。它提供了有关软件的重要信息，包括日志文件的相关信息。使用 **man** 命令显示 nginx 手册页面：

[analyst@secOps ~]$ **man nginx**

NGINX(8) BSD System Manager's Manual NGINX(8)

NAME

nginx — HTTP and reverse proxy server, mail proxy server

SYNOPSIS

nginx [-?hqTtVv] [-c file] [-g directives] [-p prefix] [-s signal]

DESCRIPTION

nginx (pronounced “engine x”) is an HTTP and reverse proxy server, as well as a mail proxy

server. It is known for its high performance, stability, rich feature set, simple configura‐

tion, and low resource consumption.

<some output omitted>

* + 1. 向下滚动页面，找到 nginx 日志记录部分。该文档清楚地表明 nginx 支持日志记录功能，在编译时就定义了日志文件的位置。

[从 nginx 手册页面摘取的部分输出]

DEBUGGING LOG

To enable a debugging log, reconfigure nginx to build with debugging:

./configure --with-debug ...

and then set the debug level of the error\_log:

error\_log /path/to/log debug;

It is also possible to enable the debugging for a particular IP address:

events {

debug\_connection 127.0.0.1;

}

* + 1. 手册页面还包含 nginx 所使用文件的相关信息。再向下滚动，显示“文件”部分下的 nginx 操作文件：

文件

%%PID\_PATH%%

Contains the process ID of nginx. The contents of this file are

not sensitive, so it can be world-readable.

%%CONF\_PATH%%

The main configuration file.

%%ERROR\_LOG\_PATH%%

Error log file.

上面的输出可帮助同学们得出结论：nginx 支持日志记录，并且可以保存到日志文件中。输出还暗示存在 nginx 的配置文件。

* + 1. 查找 nginx文件前，使用 **ps** 和 **grep** 命令确保 nginx 正在虚拟机中运行。

**注**：使用 **man** 了解 **ps** 和 **grep** 命令的更多信息。

[analyst@secOps ~]$ **ps ax | grep nginx**

415 ? Ss 0:00 nginx: master process /usr/bin/nginx -g pid /run/nginx.pid; error\_log stderr;

416 ? S 0:00 nginx: worker process

1207 pts/0 S+ 0:00 grep nginx

以上输出确认 nginx 正在运行。此外，输出还显示启动 nginx 时使用的参数。nginx 进程 ID 存储在 /run/nginx.pid 中，并且错误消息被重定向到终端。

**注**：如果 nginx 没有运行，请在提示符后输入 **sudo /usr/sbin/nginx** 以使用默认配置启动该服务。

**注**：如果需要重新启动 nginx，可以使用 **sudo pkill nginx** 命令来终止该服务。要使用上一个实验中的自定义配置启动 nginx，请运行以下命令：**sudo nginx -c custom\_server.conf**，然后打开 Web 浏览器并转至 URL：127.0.0.1:81 来测试服务器。如果同学们希望使用默认配置启动 **nginx**，可以使用以下命令：**sudo /usr/sbin/nginx**，然后打开 Web 浏览器并转至 URL：127.0.0.1。

由于没有指定日志文件的位置，因此应检查全局 nginx 配置文件以确定日志文件的位置。

* + 1. 根据设计，CyberOps Workstation 虚拟机会尽可能利用默认位置和定义。通常，/var/log 目录保留各种应用和服务的各种日志文件，而配置文件存储在 /etc 目录下。虽然 nginx 手册页面没有提供日志文件的确切位置，但它不仅确认了 nginx 支持日志记录功能，而且还暗示了配置文件的位置。由于通常可以在配置文件中自定义日志文件的位置，因此下一步应该是使用 **ls** 命令查看 /etc 下的位置，并查找 nginx 配置文件：

[analyst@secOps ~]$ **ls /etc/**

adjtime host.conf mke2fs.conf rc\_maps.cfg

apache-ant hostname mkinitcpio.conf request-key.conf

apparmor.d hosts mkinitcpio.d request-key.d

arch-release ifplugd modprobe.d resolv.conf

avahi initcpio modules-load.d resolvconf.conf

bash.bash\_logout inputrc motd rpc

bash.bashrc iproute2 mtab rsyslog.conf

binfmt.d iptables nanorc securetty

ca-certificates issue netconfig security

crypttab java-7-openjdk netctl services

dbus-1 java-8-openjdk netsniff-ng shadow

default kernel nginx shadow-

depmod.d krb5.conf nscd.conf shells

dhcpcd.conf ld.so.cache nsswitch.conf skel

dhcpcd.duid ld.so.conf ntp.conf ssh

dkms ld.so.conf.d openldap ssl

drirc libnl openvswitch sudoers

elasticsearch libpaper.d os-release sudoers.d

environment lightdm pacman.conf sudoers.pacnew

ethertypes locale.conf pacman.conf.pacnew sysctl.d

<output omitted>

* + 1. 注意上面的输出中 /etc 下的 nginx 文件夹。再次使用 ls，我们找到了大量文件，包括名为 nginx.conf 的文件。

[analyst@secOps ~]$ **ls -l /etc/nginx/**

total 48

-rw-r--r-- 1 root root 2730 Mar 21 16:02 custom\_server.conf

-rw-r--r-- 1 root root 1077 Nov 18 15:14 fastcgi.conf

-rw-r--r-- 1 root root 1007 Nov 18 15:14 fastcgi\_params

-rw-r--r-- 1 root root 2837 Nov 18 15:14 koi-utf

-rw-r--r-- 1 root root 2223 Nov 18 15:14 koi-win

-rw-r--r-- 1 root root 2743 Jan 6 15:41 mal\_server.conf

-rw-r--r-- 1 root root 3957 Nov 18 15:14 mime.types

-rw-r--r-- 1 root root 3264 Mar 22 13:34 nginx.conf

-rw-r--r-- 1 root root 3261 Oct 19 16:42 nginx.conf.working

-rw-r--r-- 1 root root 636 Nov 18 15:14 scgi\_params

-rw-r--r-- 1 root root 664 Nov 18 15:14 uwsgi\_params

-rw-r--r-- 1 root root 3610 Nov 18 15:14 win-utf

* + 1. 使用 **cat** 命令列出 /etc/nginx/nginx.conf 的内容。同学们还可以使用 **more** 或 **less** 来查看该文件，使用 **nano** 或 **SciTE** 对文件进行编辑。使用这些工具更方便浏览较长的文本文件（下面仅显示 cat 的输出）。

[analyst@secOps ~]$ **cat /etc/nginx/nginx.conf**

#user html;

worker\_processes 1;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

worker\_connections 1024;

}

<some output omitted>

**注**：以“#”开头的行表示注释，会被 **nginx** 忽略。

* + 1. 快速查看该配置文件，可以发现它是一个 nginx 配置文件。由于没有直接提及 nginx 日志文件的位置，因此很有可能 nginx 使用了默认值。按照将日志文件存储到 /var/log 下的惯例，使用 **ls** 命令列出其内容：

[analyst@secOps log]$ **ls -l /var/log/**

total 1760

-rw------- 1 root utmp 384 Mar 20 08:34 btmp

-rw------- 1 root utmp 384 Feb 13 19:50 btmp.1

-rw-r----- 1 root root 6204 Mar 22 06:01 debug

-rw-r----- 1 root root 18612 Mar 20 04:34 debug.1

-rw-r----- 1 root root 6116 Feb 24 10:25 debug.2

-rw-r----- 1 root root 6116 Feb 14 10:39 debug.3

-rw-r----- 1 root root 12232 Feb 13 14:50 debug.4

drwxr-xr-x 2 elasticsearch elasticsearch 4096 Dec 5 17:26 elasticsearch

-rw------- 1 root root 24024 Mar 22 10:03 faillog

drwxr-xr-x 2 filebeat filebeat 4096 Nov 30 17:19 filebeat

drwxr-sr-x+ 4 root systemd-journal 4096 Sep 7 2016 journal

-rw-r----- 1 root root 42990 Mar 22 10:30 kern.log

-rw-r----- 1 root root 146594 Mar 20 08:34 kern.log.1

-rw-r----- 1 root root 40170 Feb 24 16:20 kern.log.2

-rw-r----- 1 root root 41466 Feb 21 09:11 kern.log.3

-rw-r----- 1 root root 79337 Feb 13 19:50 kern.log.4

-rw-r--r-- 1 root root 292292 Dec 2 08:38 lastlog

drwx--x--x 2 root lightdm 4096 Mar 22 10:01 lightdm

-rw-r--r-- 1 analyst analyst 24464 Feb 7 13:41 logstash-tutorial.log

-rw-r----- 1 root root 36318 Mar 22 10:30 messages

-rw-r----- 1 root root 126578 Mar 20 08:34 messages.1

-rw-r----- 1 root root 33781 Feb 24 16:20 messages.2

-rw-r----- 1 root root 35077 Feb 21 09:11 messages.3

-rw-r----- 1 root root 66559 Feb 13 19:50 messages.4

drwxr-x--- 2 http log 4096 Mar 20 08:34 nginx

-rw-r--r-- 1 http root 989 Dec 2 10:30 nginx-logstash.log

drwxr-xr-x 2 root root 4096 Jul 4 2014 old

drwxr-xr-x 2 root root 4096 Mar 6 2016 openvswitch

-rw-r--r-- 1 root root 209907 Mar 20 08:37 pacman.log

-rw-r----- 1 root root 42990 Mar 22 10:30 syslog

-rw-r----- 1 root root 146594 Mar 20 08:34 syslog.1

-rw-r----- 1 root root 40170 Feb 24 16:20 syslog.2

-rw-r----- 1 root root 41466 Feb 21 09:11 syslog.3

-rw-r----- 1 root root 40936 Feb 13 19:50 syslog.4

-rw-rw-r-- 1 root utmp 353664 Mar 22 10:03 wtmp

-rw-r--r-- 1 root root 23720 Mar 22 10:03 Xorg.0.log

-rw-r--r-- 1 root root 25536 Mar 21 16:05 Xorg.0.log.old

* + 1. 如上所示，**/var/log** 目录有一个名为 **nginx** 的子目录。再次使用 **ls** 命令列出 **/var/log/nginx** 的内容。

**注**：由于 **/var/log/nginx** 属于 **http** 用户，同学们必须将 **ls** 作为 **root** 执行，只需在前面加上 **sudo** 命令即可。

[analyst@secOps log]$ **sudo ls -l /var/log/nginx**

[sudo] password for analyst:

total 20

-rw-r----- 1 http log 2990 Mar 22 11:20 access.log

-rw-r----- 1 http log 141 Feb 28 15:57 access.log.1.gz

-rw-r----- 1 http log 178 Jan 6 16:29 access.log.2.gz

-rw-r----- 1 http log 145 Dec 7 09:12 access.log.3.gz

-rw-r----- 1 http log 169 Nov 30 15:44 access.log.4.gz

<output omitted>

这些文件很可能是 nginx 使用的日志文件。进入下一部分以监控这些文件，并确认它们确实是 nginx 日志文件。

1. 实时监控日志文件

如前几部分所示，日志文件可以用许多文本表示工具来显示。虽然 **cat**、**more**、**less** 和 **nano** 可用于处理日志文件，但它们不适用于实时监控日志文件。开发人员设计出了可实时监控日志文件的各种工具。有些工具是基于文本的，有些工具则具备图形界面。本实验重点介绍 **tail**，这是一款简单却有效的工具，可在基于 Unix 的所有系统中使用。

* 1. 使用 tail 命令

**Tail** 命令显示文本文件的末尾。默认情况下，**tail** 将显示文本文件的最后十 (10) 行。

* + 1. 使用 **tail** 命令显示 **/var/log/nginx/access.log** 的末尾。

**注**：同学们必须在 tail 之前加上 **sudo** 命令，这是因为 /var/log/nginx/access.log 属于 http 用户。

**[analyst@secOps log]$** **sudo tail /var/log/nginx/access.log**

[sudo] password for analyst:

127.0.0.1 - - [21/Mar/2017:15:32:32 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:34 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:44 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:11:20:27 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:26 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:50 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:53 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:13:01:55 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

* + 1. 使用 **–n** 选项指定 **tail** 应显示文件末尾之前的多少行。

[analyst@secOps log]$ **sudo tail –n 5 /var/log/nginx/access.log**

[sudo] password for analyst:

127.0.0.1 - - [22/Mar/2017:11:20:27 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:26 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:50 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:53 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:13:01:55 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

* + 1. 可以将 **tail** 命令与 **-f** 选项配合使用，以实时监控 nginx access.log。**-f** 是 follow 的缩写，用于告知 **tail** 连续显示给定文本文件的末尾。在终端窗口中，发出 **tail** 与 **–f** 选项：

**[analyst@secOps log]$** **sudo tail -f /var/log/nginx/access.log**

[sudo] password for analyst:

127.0.0.1 - - [21/Mar/2017:15:32:32 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:34 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [21/Mar/2017:15:32:44 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:11:20:27 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:26 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:12:49:50 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

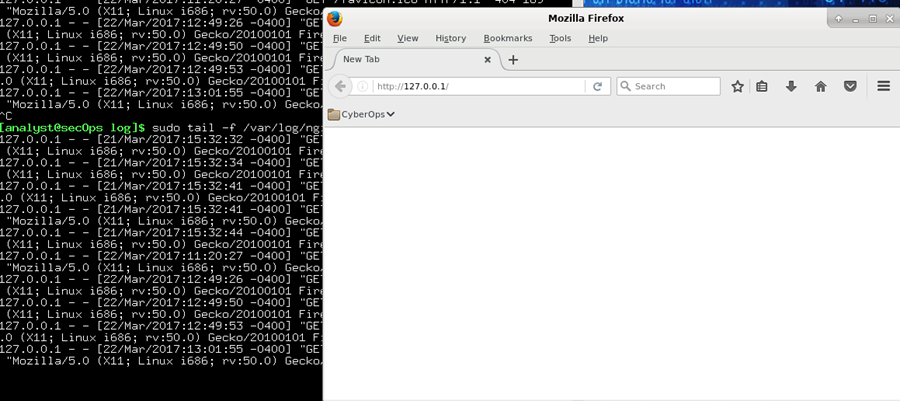
127.0.0.1 - - [22/Mar/2017:12:49:53 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

127.0.0.1 - - [22/Mar/2017:13:01:55 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

像之前一样，**tail** 显示文件的最后 10 行。但请注意，**tail** 在显示这些行后并不退出；命令提示符不可见，表示 **tail** 仍在运行。

* + 1. 如果 **tail** 仍在终端窗口中运行，请点击停靠栏上的 Web 浏览器图标以打开 Web 浏览器窗口。重新调整网络浏览器视窗的大小，确保可以看到仍在运行 **tail** 的终端窗口的底部。

**注**：在下面的截图中，在运行 **tail** 的终端窗口中 Enter 键被按了几次。这仅仅是为了呈现可视化效果，因为在和 **–f** 一起运行时 **tail** 不处理任何输入。额外的空行便于检测新条目，因为新条目显示在终端窗口的底部。



* + 1. 在 Web 浏览器地址栏中，输入 **127.0.0.1** 并按 Enter。这是虚拟机本身的地址，它告诉浏览器连接到在本地计算机上运行的 Web 服务器。新条目应记录在 /var/log/nginx/access.log 文件中。刷新网页以查看日志中添加的新条目。

127.0.0.1 - - [23/Mar/2017:9:48:36 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"

由于 **tail** 仍在运行，因此它应在终端窗口的底部显示新条目。除了时间戳之外，同学们的条目应看起来和上面的条目相似。

**注**：Firefox 将页面存储在缓存中以备将来使用。如果某个页面已存储在缓存中，可强制 Firefox 忽略缓存并提交 Web 请求，然后按 **<CTRL+SHIFT+R>** 加载该页面。

* + 1. 由于日志文件是由 nginx 更新的，我们可以肯定 /var/log/acess.log 实际上是 nginx 正在使用的日志文件。
    2. 输入 **Ctrl + C** 结束 tail 监控会话。
  1. 额外工具：Journalctl

CyberOps Workstation 虚拟机基于 Arch Linux。Arch Linux 被归类为 Linux 发行版，其设计理念是轻型、简约、简单。作为这种设计理念的一部分，Arch Linux 使用 systemd 作为初始化系统。在 Linux 中，初始化进程是计算机引导时加载的第一个进程。无论是直接还是间接关系，初始化都是系统上所有运行的进程的父进程。它在引导时由内核启动，并继续运行直到计算机关闭。通常情况下，初始化的进程 ID 为 1。

初始化系统包含一套规则和惯例，用于管理特定 Linux 系统中的用户空间的创建和提供方式。初始化系统还指定适用于整个系统的参数，如全局配置文件、日志记录结构和服务管理。

systemd 是一种现代初始化系统，旨在统一所有 Linux 发行版中的 Linux 配置和服务行为，并且越来越广泛地为主要 Linux 发行版所采用。Arch Linux 依赖于 systemd 来实现初始化功能。CyberOps Workstation 虚拟机也使用 systemd。

system-journald（或简称 journald）是 systemd 的事件日志记录服务程序，它使用 AOF 二进制文件作为日志文件。请注意，journald 不妨碍 syslog 和 rsyslog 等其他日志记录系统的使用。

此部分简要概述 journalctl，journald 实用程序用于查看和实时监控日志。

* + 1. 在 CyberOps Workstation 虚拟机的终端窗口中，发出 journalctl 命令，不要包含显示所有日志条目（可能相当多）的选项：

**[analyst@secOps ~]$ journalctl**

Hint: You are currently not seeing messages from other users and the system.

Users in groups 'adm', 'systemd-journal', 'wheel' can see all messages.

Pass -q to turn off this notice.

-- Logs begin at Fri 2014-09-26 14:13:12 EDT, end at Fri 2017-03-31 09:54:58 EDT

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Paths.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Paths.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Timers.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Timers.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Sockets.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Sockets.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Basic System.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Basic System.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Default.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Default.

Sep 26 14:13:12 dataAnalyzer systemd[1087]: Startup finished in 18ms.

Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopping Default.

<some output omitted>

输出开始于类似以下内容的一行，标记系统开始记录日志的时间戳。请注意，时间戳因系统而异。

-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:12:19 EDT.–-

journalctl 包括许多功能，如页面滚动、带颜色编码的消息等。使用键盘上/下箭头键可上下滚动输出，一次滚动一行。使用键盘左/右箭头键可左右滚动，并显示跨越终端窗口边界的日志条目。在输出中，按 **<ENTER>** 显示下一行，按空格键显示下一页。按 **q** 键退出 journalctl。

请注意 journalctl 提供的提示消息：

Hint: You are currently not seeing messages from other users and the system.

Users in groups 'adm', 'systemd-journal', 'wheel' can see all messages.

Pass -q to turn off this notice.

此消息提醒同学们，由于分析师是普通用户而非 adm、systemd-journal 或 wheel 组的成员，因此 journalctl 不会显示所有日志条目。它还指出，运行带 –q 选项的 journalctl 不会提示本消息

如何运行 **journalctl** 并查看所有日志条目？

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. **Journalctl** 包括有助于过滤输出的选项。使用 **–b** 选项显示与引导相关的日志条目：

[analyst@secOps ~]$ **sudo journalctl -b**

-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:18:04 EDT.--

Mar 31 05:54:43 secOps systemd-journald[169]: Time spent on flushing to /var is 849us for 0 entries.

Mar 31 05:54:43 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andyrtr) (gcc version 6.2.1 20160830 (GCC) ) #1 SMP PREEM

Mar 31 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'

Mar 31 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'

Mar 31 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'

Mar 31 05:54:43 secOps kernel: x86/fpu: xstate\_offset[2]: 576, xstate\_sizes[2]: 256

Mar 31 05:54:43 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.

Mar 31 05:54:43 secOps kernel: x86/fpu: Using 'eager' FPU context switches.

Mar 31 05:54:43 secOps kernel: e820: BIOS-provided physical RAM map:

Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable

Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved

Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved

Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x000000007ffeffff] usable

<some output omitted>

* + 1. 要查看与最后一次引导相关的条目，请在上面的命令中添加 **-1** 选项。要查看与最后两次引导相关的条目，请添加 **-2** 选项。

[analyst@secOps ~]$ **sudo journalctl –b -2**

-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:21:03 EDT.--

Mar 22 09:35:11 secOps systemd-journald[181]: Time spent on flushing to /var is 4.204ms for 0 entries.

Mar 22 9:35:11 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andyrtr) (gcc version 6.2.1 20160830 (GCC) ) #1 SMP PREEM

Mar 22 9:35:11 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'

Mar 22 9:35:11 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'

Mar 22 09:35:11 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'

Mar 22 09:35:11 secOps kernel: x86/fpu: xstate\_offset[2]: 576, xstate\_sizes[2]: 256

Mar 22 9:35:11 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.

Mar 22 9:35:11 secOps kernel: x86/fpu: Using 'eager' FPU context switches.

Mar 22 9:35:11 secOps kernel: e820: BIOS-provided physical RAM map:

Mar 22 9:35:11 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable

Mar 22 9:35:11 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved

Mar 22 9:35:11 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved

Mar 22 9:35:11 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x000000007ffeffff] usable

Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x000000007fff0000-0x000000007fffffff] ACPI data

Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved

Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved

<some output omitted>

* + 1. 使用 **--list-boots** 选项列出之前的引导：

[analyst@secOps ~]$ **sudo journalctl –-list-boots**

-144 fbef03a1b59c40429f3e083613ab775a Fri 2014-09-26 13:22:51 EDT—Fri 2014-09-26 14:05:00 EDT

-143 69ebae646d6b41f0b3de9401cb3aa591 Fri 2014-09-26 14:05:07 EDT—Fri 2014-09-26 20:35:29 EDT

-142 73a305f65dea41e787b164411dfc6750 Fri 2014-09-26 20:35:34 EDT—Fri 2014-09-26 20:52:22 EDT

-141 48a113d5d2f44979a849c9c0d9ecdfa2 Fri 2014-09-26 20:52:33 EDT—Fri 2014-09-26 21:08:35 EDT

-140 002af74c3fc44008a882384f546c438d Fri 2014-09-26 21:08:45 EDT—Fri 2014-09-26 21:16:39 EDT

-139 f3ca1d06495c4e26b367e6867f03374c Fri 2014-09-26 21:16:47 EDT—Fri 2014-09-26 21:50:19 EDT

-138 bd232f288e544a79aa3bc444e02185a8 Fri 2014-09-26 21:50:28 EDT—Fri 2014-09-26 22:33:13 EDT

-137 2097c11f249c431aa8ad8da31a5b26d1 Fri 2014-09-26 22:40:39 EDT—Fri 2014-09-26 23:55:46 EDT

-136 b24d5e718a724b18b352e9b2daed3db6 Sat 2014-09-27 10:57:32 EDT—Sat 2014-09-27 14:26:43 EDT

-135 5a189fc68352484a8b40cd719ff7dd41 Sat 2014-09-27 19:44:23 EDT—Sat 2014-09-27 22:50:24 EDT

-134 d0be08c1f26642a1a20bb70bfc7b722c Mon 2014-09-29 09:17:14 EDT—Mon 2014-09-29 12:12:10 EDT

-133 b00b0d4c07464071b0d3cac4eb79dda3 Mon 2014-09-29 12:39:12 EDT—Mon 2014-09-29 13:24:38 EDT

<some output omitted>

* + 1. 使用 **--since “<time range>”** 指定显示日志条目的时间范围。下面的两个命令分别用于显示最后两小时和最后一天生成的所有日志条目：

[analyst@secOps ~]$ **sudo journalctl –-since "2 hours ago"**

-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:28:29 EDT.--

Mar 31 09:54:45 secOps kernel: 00:00:00.008577 main 5.1.10 r112026 started.Verbose level = 0

Mar 31 09:54:45 secOps systemd[1]: Time has been changed

Mar 31 09:54:45 secOps systemd[1]: Started Rotate log files.

Mar 31 09:54:45 secOps ovsdb-server[263]: 2017-03-31T13:54:45Z|00001|ovsdb\_server|INFO|ovsdb-server (Open vSwitch) 2.6.1

Mar 31 09:54:45 secOps ovsdb-server[263]: ovs|00001|ovsdb\_server|INFO|ovsdb-server (Open vSwitch) 2.6.1

Mar 31 09:54:45 secOps kernel: openvswitch: Open vSwitch switching datapath

Mar 31 09:54:45 secOps systemd[1]: Started Open vSwitch Daemon.

Mar 31 09:54:45 secOps dhcpcd[279]: enp0s3: soliciting an IPv6 router

Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-31T13:54:45Z|00001|ovs\_numa|INFO|Discovered 1 CPU cores on NUMA node 0

Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-31T13:54:45Z|00002|ovs\_numa|INFO|Discovered 1 NUMA nodes and 1 CPU cores

Mar 31 09:54:45 secOps ovs-vswitchd[319]: ovs|00001|ovs\_numa|INFO|Discovered 1 CPU cores on NUMA node 0

Mar 31 09:54:45 secOps ovs-vswitchd[319]: ovs|00002|ovs\_numa|INFO|Discovered 1 NUMA nodes and 1 CPU cores

Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-31T13:54:45Z|00003|reconnect|INFO|unix:/run/openvswitch/db.sock: connecting..

Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-31T13:54:45Z|00004|reconnect|INFO|unix:/run/openvswitch/db.sock: connected

Mar 31 09:54:45 secOps ovs-vswitchd[319]: ovs|00003|reconnect|INFO|unix:/run/openvswitch/db.sock: connecting...

Mar 31 09:54:45 secOps ovs-vswitchd[319]: ovs|00004|reconnect|INFO|unix:/run/openvswitch/db.sock: connected

Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-31T13:54:45Z|00005|ovsdb\_idl|WARN|Interface table in Open\_vSwitch database la

Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-31T13:54:45Z|00006|ovsdb\_idl|WARN|Mirror table in Open\_vSwitch database lacks

<some output omitted>

[analyst@secOps ~]$ **sudo journalctl –-since "1 day ago"**

-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:26:48 EDT.--

Mar 30 05:54:43 secOps systemd-journald[169]: Time spent on flushing to /var is 849us for 0 entries.

Mar 30 05:54:43 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andyrtr) (gcc version 6.2.1 20160830 (GCC) ) #1 SMP PREEM

Mar 30 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'

Mar 30 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'

Mar 30 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'

Mar 30 05:54:43 secOps kernel: x86/fpu: xstate\_offset[2]: 576, xstate\_sizes[2]: 256

Mar 31 05:54:43 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.

Mar 30 05:54:43 secOps kernel: x86/fpu: Using 'eager' FPU context switches.

Mar 30 05:54:43 secOps kernel: e820: BIOS-provided physical RAM map:

Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable

Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved

Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved

<some output omitted>

* + 1. journalctl 还允许使用 –u 选项显示与特定服务相关的日志条目。下面的命令用于显示与 nginx 相关的日志条目：

[analyst@secOps ~]$ **sudo journalctl –u nginx.service**

-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:30:39 EDT.--

Oct 19 16:47:57 secOps systemd[1]: Starting A high performance web server and a reverse proxy server...

Oct 19 16:47:57 secOps nginx[21058]: 2016/10/19 16:47:57 [warn] 21058#21058: conflicting server name "localhost" on 0.0.0.0:80,

Oct 19 16:47:57 secOps systemd[1]: nginx.service: PID file /run/nginx.pid not readable (yet?) after start: No such file or dire

Oct 19 16:47:57 secOps systemd[1]: Started A high performance web server and a reverse proxy server.

Oct 19 17:40:09 secOps nginx[21058]: 2016/10/19 17:40:09 [error] 21060#21060: \*1 open() "/usr/share/nginx/html/favicon.ico" fai

Oct 19 17:40:09 secOps nginx[21058]: 2016/10/19 17:40:09 [error] 21060#21060: \*1 open() "/usr/share/nginx/html/favicon.ico" fai

Oct 19 17:41:21 secOps nginx[21058]: 2016/10/19 17:41:21 [error] 21060#21060: \*2 open() "/usr/share/nginx/html/favicon.ico" fai

Oct 19 17:41:21 secOps nginx[21058]: 2016/10/19 17:41:21 [error] 21060#21060: \*2 open() "/usr/share/nginx/html/favicon.ico" fai

Oct 19 18:36:33 secOps systemd[1]: Stopping A high performance web server and a reverse proxy server...

Oct 19 18:36:33 secOps systemd[1]: Stopped A high performance web server and a reverse proxy server.

-- Reboot --

Oct 19 18:36:49 secOps systemd[1]: Starting A high performance web server and a reverse proxy server...

Oct 19 18:36:49 secOps nginx[399]: 2016/10/19 18:36:49 [warn] 399#399: conflicting server name "localhost" on 0.0.0.0:80, ignor

Oct 19 18:36:49 secOps systemd[1]: nginx.service: PID file /run/nginx.pid not readable (yet?) after start: No such file or dire

Oct 19 18:36:49 secOps systemd[1]: Started A high performance web server and a reverse proxy server.

<some output omitted>

**注**：作为 systemd 的一部分，服务被描述为单元。大多数服务安装包在安装过程中会创建并启用单元。

* + 1. 与 **tail –f** 类似，**journalctl** 也支持实时监控。使用 –f 选项指示 journalctl *跟踪*特定日志。按 **Ctrl + C** 退出。

[analyst@secOps ~]$ **sudo journalctl -f**

[sudo] password for analyst:

-- Logs begin at Fri 2014-09-26 13:22:51 EDT.--

Mar 31 10:34:15 secOps filebeat[222]: 2017/03/31 14:34:15.077058 logp.go:232: INFO No non-zero metrics in the last 30s

Mar 31 10:34:40 secOps sudo[821]: pam\_unix(sudo:session): session closed for user root

Mar 31 10:34:45 secOps filebeat[222]: 2017/03/31 14:34:45.076057 logp.go:232: INFO No non-zero metrics in the last 30s

Mar 31 10:35:15 secOps filebeat[222]: 2017/03/31 14:35:15.076118 logp.go:232: INFO No non-zero metrics in the last 30s

Mar 31 10:35:45 secOps filebeat[222]: 2017/03/31 14:35:45.076924 logp.go:232: INFO No non-zero metrics in the last 30s

Mar 31 10:36:15 secOps filebeat[222]: 2017/03/31 14:36:15.076060 logp.go:232: INFO No non-zero metrics in the last 30s

Mar 31 10:36:45 secOps filebeat[222]: 2017/03/31 14:36:45.076122 logp.go:232: INFO No non-zero metrics in the last 30s

Mar 31 10:37:15 secOps filebeat[222]: 2017/03/31 14:37:15.076801 logp.go:232: INFO No non-zero metrics in the last 30s

Mar 31 10:37:30 secOps sudo[842]: analyst : TTY=pts/0 ; PWD=/home/analyst ; USER=root ; COMMAND=/usr/bin/journalctl -f

Mar 31 10:37:31 secOps sudo[842]: pam\_unix(sudo:session): session opened for user root by (uid=0)

<some output omitted>

* + 1. journalctl 还支持混合多个选项以实现所需的过滤条件组合。下面的命令用于实时监控 nginx 系统事件。

[analyst@secOps ~]$ **sudo journalctl -u nginx.service -f**

-- Logs begin at Fri 2014-09-26 13:22:51 EDT.--

Mar 23 10:08:41 secOps systemd[1]: Stopping A high performance web server and a reverse proxy server...

Mar 23 10:08:41 secOps systemd[1]: Stopped A high performance web server and a reverse proxy server.

-- Reboot --

Mar 29 11:28:06 secOps systemd[1]: Starting A high performance web server and a reverse proxy server...

Mar 29 11:28:06 secOps systemd[1]: nginx.service: PID file /run/nginx.pid not readable (yet?) after start: No such file or directory

Mar 29 11:28:06 secOps systemd[1]: Started A high performance web server and a reverse proxy server.

Mar 29 11:31:45 secOps systemd[1]: Stopping A high performance web server and a reverse proxy server...

Mar 29 11:31:45 secOps systemd[1]: Stopped A high performance web server and a reverse proxy server.

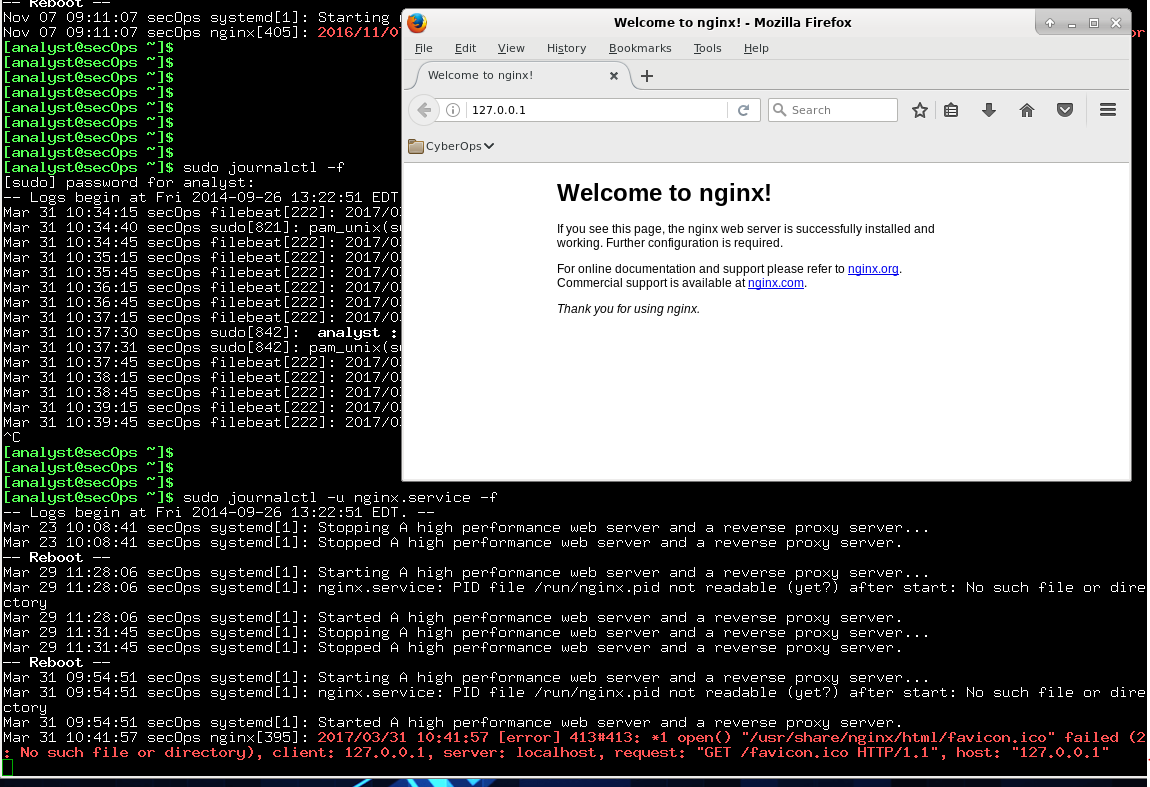
-- Reboot --

Mar 31 09:54:51 secOps systemd[1]: Starting A high performance web server and a reverse proxy server...

Mar 31 09:54:51 secOps systemd[1]: nginx.service: PID file /run/nginx.pid not readable (yet?) after start: No such file or directory

Mar 31 09:54:51 secOps systemd[1]: Started A high performance web server and a reverse proxy server.

* + 1. 使上面的命令保持运行时，打开新的 Web 浏览器窗口并在地址栏中键入 127.0.0.1（默认配置）或 127.0.0.1:81 (custom\_server.conf)。journalctl 应实时显示关于缺失 favicon.ico 文件的错误。



1. 思考

日志文件对故障排除极其重要。

日志文件位置遵循惯例，但最终还是由开发人员进行选择。

通常情况下，日志文件的信息（位置、文件名等）都包含在文档中。如果文档没有提供日志文件的有用信息，则应使用 Web 研究和系统调查来查找日志文件信息。

时钟应始终同步，以确保所有系统的时间正确。如果未正确设置时钟，则很难追溯事件。

了解具体事件发生的时间很重要。除此之外，系统往往会同时分析来自不同来源的事件。