

Laboratório - Localização de Arquivos de Log

Objetivos

Neste laboratório, você vai se familiarizar com a localização e manipulação de arquivos de log do Linux.

Parte 1: Visão geral do arquivo de log

Parte 2: Localizando arquivos de log em sistemas desconhecidos

Parte 3: Monitoramento de Arquivos de Log em Tempo Real

Recursos necessários

- Máquina virtual CyberOps Workstation

Instruções

Parte 1: Visão geral do arquivo de log

Arquivos de log (também arquivos de log ortográficos), são arquivos usados por computadores para registrar eventos. Programas de software, processos em segundo plano, serviços ou transações entre serviços, incluindo o próprio sistema operacional, podem gerar tais eventos. Os arquivos de log dependem do aplicativo que os gera. Cabe ao desenvolvedor do aplicativo estar em conformidade com a convenção do arquivo de log. A documentação do software deve incluir informações sobre seus arquivos de log.

Etapa 1: Exemplo de arquivo de log do servidor Web

Como os arquivos de log são essencialmente uma maneira de rastrear eventos específicos, o tipo de informações armazenadas varia dependendo do aplicativo ou serviços que geram os eventos.

- a. Considere a entrada de log única abaixo. Foi gerado pelo Apache, um servidor web popular.

```
[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
```

A entrada de log única acima representa um evento web gravado pelo Apache. Algumas informações são importantes nas transações da web, incluindo endereço IP do cliente, hora e detalhes da transação. A entrada acima pode ser dividida em cinco partes principais:

Carimbo de data/hora: Esta parte registra quando o evento ocorreu. É muito importante que o relógio do servidor esteja corretamente sincronizado, pois permite cruzar com precisão os eventos e rastrear.

Tipo: Este é o tipo de evento. Neste caso, foi um erro.

PID: Contém informações sobre o ID do processo usado pelo Apache no momento.

Cliente: Isto registra o endereço IP do cliente solicitante.

Descrição: contém uma descrição do evento.

Com base na entrada de log acima, descreva o que aconteceu.

Use o comando **cat** abaixo para listar um arquivo de log de exemplo do servidor Web. O arquivo de exemplo está localizado em `/var/log`:

```
[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>
```

A saída acima ainda é considerada uma transação web? Explique por que a saída do comando **cat** está em um formato diferente da entrada única mostrada no item (a).

Etapa 2: Exemplo de arquivo de log do sistema operacional

Qualquer software pode manter arquivos de log, incluindo o próprio sistema operacional. Convencionalmente, o Linux usa o diretório `/var/log` para armazenar vários arquivos de log, incluindo logs do sistema operacional. Os sistemas operacionais modernos são peças complexas de software e, portanto, usam vários arquivos diferentes para registrar eventos. Esta seção dá uma olhada rápida no arquivo `/var/log/messages`.

- a. Armazenado em `/var/log`, o arquivo de mensagens armazena vários eventos do sistema. A conexão de uma nova unidade USB, uma placa de rede se tornando disponível e muitas tentativas de login root perdidas, são alguns exemplos de eventos registrados no arquivo `/var/log/messages`. Use o comando **more** para exibir o conteúdo do arquivo `/var/log/messages`. Ao contrário do comando **cat**, **more** permite uma navegação em ritmo através do arquivo. Pressione **ENTER** para avançar linha a linha ou **ESPAÇO** para avançar uma página inteira. Pressione **q** ou **CTRL + C** para abortar e sair **mais**.

Nota: o comando **sudo** é necessário porque o arquivo de mensagens pertence ao usuário root.

```
[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
```

Laboratório - Localização de Arquivos de Log

```
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@andyrt) (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.
<alguma saída omitida>
```

Observe que os eventos listados acima são muito diferentes dos eventos do servidor web. Como o próprio sistema operacional está gerando esse log, todos os eventos registrados estão em relação ao próprio sistema operacional.

- b. Se necessário, digite **Ctrl + C** para sair do comando anterior.
- c. Os arquivos de log são muito importantes para a solução de problemas. Suponha que um usuário desse sistema específico informou que todas as operações de rede foram lentas em torno de 4:20am em 19 de maio.

Você pode encontrar evidências disso nas entradas de log mostradas acima? Em caso afirmativo, em que linhas? Explique.

Parte 2: Localizando arquivos de log em sistemas desconhecidos

A VM CyberOps Workstation inclui nginx, um servidor web leve. Esta seção mostrará como localizar e exibir logs nginx usando a VM CyberOps Workstation.

Observação: o nginx foi instalado na VM CyberOps Workstation com suas configurações padrão. Com as configurações padrão, seu arquivo de configuração global está localizado em `/etc/nginx/nginx.conf`, seu arquivo de log de acesso está em `/var/log/nginx/access.log` e os erros são redirecionados para a janela do terminal. No entanto, é comum que um analista de segurança trabalhe em computadores nos quais os detalhes de instalação da ferramenta e dos serviços são desconhecidos. Esta seção descreve o processo de localizar esses arquivos descritos para o nginx, mas não está de modo algum completo. No entanto, deve ser um bom exercício sobre a localização e exibição de arquivos de log em sistemas desconhecidos.

Laboratório - Localização de Arquivos de Log

- a. Ao trabalhar com um novo software, o primeiro passo é olhar para a documentação. Ele fornece informações importantes sobre o software, incluindo informações sobre seus arquivos de log. Use o comando **man** para exibir a página de manual do nginx:

```
[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 [-?hqtTv] [-c file] [-g directives] [-p prefix] [-s signal]
```

DESCRIÇÃO

nginx (pronunciado “engine x”) é um servidor HTTP e proxy reverso, bem como um proxy de e-mail

servidor remoto. É conhecido por seu alto desempenho, estabilidade, conjunto de recursos ricos, configuração simples-

e baixo consumo de recursos.

< alguma saída omitida >

- b. Role a página para baixo para localizar a seção de log do nginx. A documentação deixa claro que o nginx suporta o registro em log, com a localização de seus arquivos de log definidos no momento da compilação.

[SAÍDA PARCIAL EXTRAÍDA DA PÁGINA DE MANUAL DO NGINX]

REGISTRO DE DEPURAÇÃO

Para habilitar um log de depuração, reconfigure o nginx para compilar com depuração:

```
./configure --with-debug...
```

e, em seguida, defina o nível de depuração do error_log:

```
error_log /caminho/para/log debug;
```

Também é possível habilitar a depuração para um endereço IP específico:

```
events {
    debug_connection 127.0.0.1;
}
```

- c. A página do manual também contém informações sobre os arquivos usados pelo nginx. Role para baixo para exibir os arquivos operacionais do nginx na seção Arquivos:

ARQUIVOS

%%PID_PATH%%

Contém o ID do processo do nginx. O conteúdo deste arquivo é não sensível, para que possa ser legível em todo o mundo.

%%CONF_PATH%%

O arquivo de configuração principal.

Laboratório - Localização de Arquivos de Log

%%ERROR_LOG_PATH%%

Arquivo de log de erro.

As saídas acima ajudam você a concluir que o nginx suporta log e que ele pode salvar em arquivos de log. A saída também sugere a existência de um arquivo de configuração para o nginx.

- d. Antes de procurar arquivos nginx, use os comandos **ps** e **grep** para garantir que o nginx esteja sendo executado na VM.

Nota: Use **man** para saber mais sobre os comandos **ps** e **grep**

```
[analista @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
```

A saída acima confirma que o nginx está sendo executado. Além disso, a saída também exibe os parâmetros usados quando o nginx foi iniciado. O ID do processo nginx está sendo armazenado em /run/nginx.pid e as mensagens de erro estão sendo redirecionadas para o terminal.

Nota: Se o nginx não estiver em execução, insira o **sudo /usr/sbin/nginx** no prompt para iniciar o serviço usando a configuração padrão.

Nota: Se você precisar reiniciar o nginx, você pode encerrar o serviço usando o comando **sudo pkill nginx**. Para iniciar o nginx com a configuração personalizada de um laboratório anterior, execute o seguinte comando: **sudo nginx -c custom_server.conf**, e teste o servidor abrindo um navegador da Web e indo para URL: 127.0.0.1:8080. Se você deseja iniciar o **nginx** com uma configuração padrão, você pode iniciá-lo com o comando: **sudo /usr/sbin/nginx**, e abrir um navegador web e ir para URL: 127.0.0.1.

Como o local para os arquivos de log não foi especificado, o arquivo de configuração global nginx deve ser verificado quanto à localização dos arquivos de log.

- e. Por design, a VM CyberOps Workstation utiliza, tanto quanto possível, locais e definições padrão. Convencionalmente, o diretório /var/log contém vários arquivos de log para vários aplicativos e serviços, enquanto os arquivos de configuração são armazenados no diretório /etc. Embora a página do manual do nginx não forneça uma localização exata para seus arquivos de log, ela não só confirmou que o nginx suporta log, mas também sugeriu a localização de um arquivo de configuração. Como os locais dos arquivos de log geralmente podem ser personalizados em arquivos de configuração, uma próxima etapa lógica é usar o comando **ls** para procurar em /etc e procurar um arquivo de configuração nginx:

```
[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
```

Laboratório - Localização de Arquivos de Log

```
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>
```

- f. Observe a pasta `nginx` sob `/etc` na saída acima. Usando `ls` novamente, encontramos um número de arquivos, incluindo um chamado `nginx.conf`.

```
[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
```

- g. Use o comando `cat` para listar o conteúdo de `/etc/nginx/nginx.conf`. Você também pode usar **more** ou **less** para visualizar o arquivo e **nano** ou **SCite** para editá-lo. Essas ferramentas facilitam a navegação por arquivos de texto longos (somente a saída do `cat` é exibida abaixo).

```
[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;
}

<alguma saída omitida >
```

Nota: As linhas que começam com `#` são comentários e são ignoradas pelo `nginx`.

- h. Uma rápida olhada no arquivo de configuração revela que ele é um arquivo de configuração `nginx`. Como não há nenhuma menção direta à localização dos arquivos de log do `nginx`, é muito provável que o `nginx` esteja usando valores padrão para ele. Seguindo a convenção de armazenamento de arquivos de log em `/var/log`, use o comando `ls` para listar seu conteúdo:

```
[analyst@secOps ~]$ ls -l /var/log/
total 5708
```

Laboratório - Localização de Arquivos de Log

```
-rw-r----- 1 root log 188962 Apr 19 10:35 auth.log
-rw-rw---- 1 root utmp 384 Apr 19 10:05 bttmp
-rw-rw---- 1 root utmp 1536 Mar 22 08:50 bttmp.1
-rw-r----- 1 root log 849038 Apr 19 10:05 daemon.log
-rw-r----- 1 root log 4416 Apr 19 09:45 errors.log
-rw-r----- 1 root log 1819814 Apr 19 10:05 everything.log
-rw----- 1 root root 32032 Apr 19 10:05 faillog
drwxr-sr-x+ 4 root systemd-journal 4096 Mar 20 15:28 journal
-rw-r----- 1 root log 927701 Apr 19 09:45 kernel.log
-rw-rw-r-- 1 root utmp 292292 Mar 26 11:03 lastlog
drwx--x--x 2 root lightdm 4096 Apr 19 09:45 lightdm
-rw-r--r-- 1 analyst analyst 24464 Apr 19 10:05 logstash-tutorial.log
-rw-r----- 1 root log 1673153 Apr 19 10:05 messages
drwxr-xr-x 2 root root 4096 Apr 19 10:28 nginx
-rw-r--r-- 1 http root 989 Apr 19 10:05 nginx-logstash.log
drwxr-xr-x 2 root root 4096 Jan 5 14:17 old
-rw-r--r-- 1 root root 97655 Apr 17 12:52 pacman.log
drwxr-xr-x 2 snort 4096 Mar 26 11:03 snort
-rw-r----- 1 root log 563 Apr 19 09:45 syslog.log
-rw----- 1 root root 64064 Mar 26 11:03 tallylog
-rw-r----- 1 root log 216 Apr 17 13:04 user.log
-rw-rw-r-- 1 root utmp 70272 Apr 19 09:45 wtmp
-rw-r--r-- 1 root root 24756 Apr 19 09:45 Xorg.0.log
-rw-r--r-- 1 root root 25585 Apr 17 14:43 Xorg.0.log.old
```

- i. Como mostrado acima, o diretório **/var/log** tem um subdiretório chamado **nginx**. Use o comando **ls** novamente para listar o conteúdo de **/var/log/nginx**.

Observação: Como o **/var/log/nginx** pertence ao usuário **http**, você deve executar **ls** como **root**, precedendo-o com o comando **sudo**.

```
[analyst@secOps ~]$ sudo ls -l /var/log/nginx
[sudo] password for analyst:
total 16
-rw-r----- 1 http log 0 May 18 17:53 access.log
-rw-r----- 1 http log 175 May 6 09:42 access.log.1.gz
-rw-r----- 1 http log 593 May 5 16:58 access.log.2.gz
-rw-r----- 1 http log 193 Jul 19 2018 access.log.3.gz
-rw-r----- 1 http log 425 Apr 19 2018 access.log.4.gz
```

Estes são muito provavelmente os arquivos de log em uso pelo **nginx**. Vá para a próxima seção para monitorar esses arquivos e obter a confirmação de que eles são realmente arquivos de log **nginx**.

Nota: Sua saída pode ser diferente. Os arquivos de log **.GZ** acima foram gerados por um serviço de rotação de log. Os sistemas Linux geralmente implementam um serviço para girar logs, garantindo que os arquivos de log individuais não se tornem muito grandes. O serviço de rotação de log pega o arquivo de log mais recente, o comprime e o salva com um nome diferente (**access.log.1.gz**, **access.log.2.gz**, etc.). Um novo arquivo de log principal vazio é então criado e usado para armazenar as entradas de log mais recentes.

Parte 3: Monitorando arquivos de log em tempo real

Como visto nas seções anteriores, os arquivos de log podem ser exibidos com muitas ferramentas de apresentação de texto. Embora **cat**, **more**, **lesse** **nano** possam ser usados para trabalhar com arquivos de

log, eles não são adequados para monitoramento de arquivos de log em tempo real. Os desenvolvedores projetaram várias ferramentas que permitem o monitoramento de arquivos de log em tempo real. Algumas ferramentas são baseadas em texto, enquanto outras têm uma interface gráfica. Este laboratório se concentra na **tail**, uma ferramenta simples mas eficiente, disponível em praticamente todos os sistemas baseados em UNIX.

Etapa 1: Usando o comando tail

O comando **tail** exibe o final de um arquivo de texto. Por padrão, o **tail** exibirá as últimas dez (10) linhas de um arquivo de texto.

Observação: Se você não vir nenhuma entrada de log, navegue até 127.0.0.1 em um navegador da Web e atualize a página algumas vezes.

- a. Use o comando **tail** para exibir o final do **/var/log/nginx/access.log**.

```
[analyst@secOps ~]$ sudo tail /var/log/nginx/access.log
[sudo] password for analyst:
127.0.0.1 - - [21/May/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/May/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/May/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/May/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/May/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/May/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/May/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/May/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/May/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/May/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"
[analyst@secOps ~]$
```

Observação: Se você não vir nenhuma entrada de log, navegue até 127.0.0.1 em um navegador da Web e atualize a página algumas vezes.

- b. Use a opção **-n** para especificar quantas linhas do final de um arquivo, a **tail** deve exibir.

```
[analyst@secOps ~]$ sudo tail -n 5 /var/log/nginx/access.log
127.0.0.1 - - [22/May/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/May/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/May/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/May/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/May/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"
```


Laboratório - Localização de Arquivos de Log

```
[analyst@secOps ~]$
```

- c. Você pode usar o comando **tail** com a opção **-f** para monitorar o `nginx access.log` em tempo real. Abreviação para seguir, **-f** diz à **tail** para exibir continuamente o final de um determinado arquivo de texto. Em uma janela de terminal, execute o **tail** com a opção **-f**:

```
[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"
```

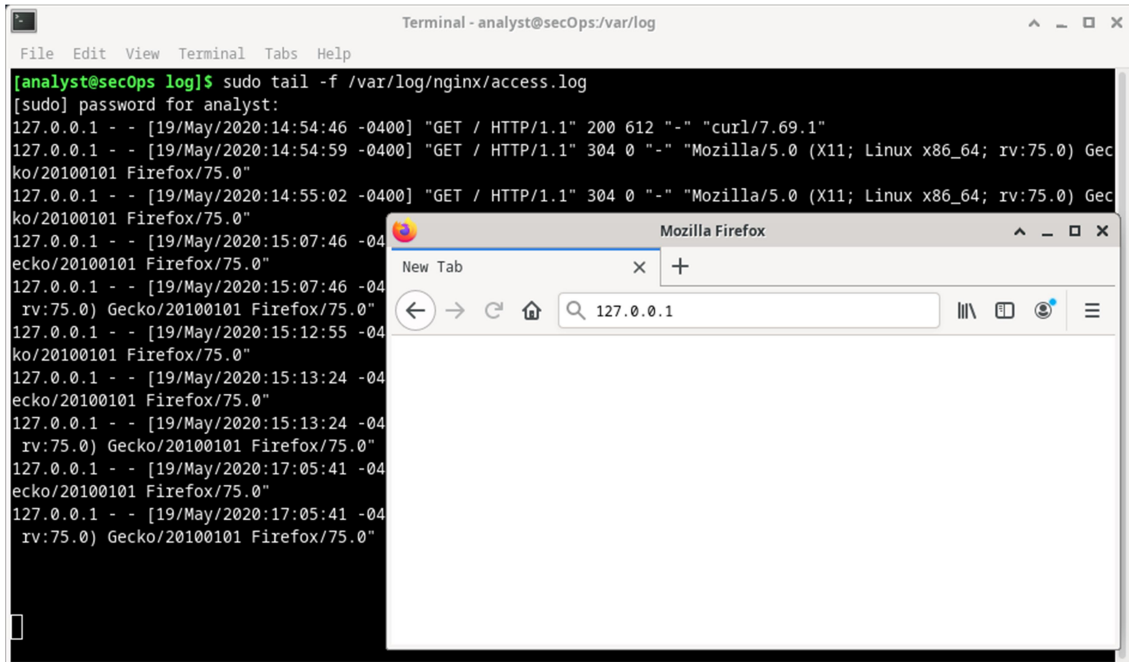
Como antes, a **tail** exibe as últimas 10 linhas do arquivo. No entanto, observe que a **tail** não sai depois de exibir as linhas; o prompt de comando não está visível, indicando que a **tail** ainda está em execução.

Observação: Seu arquivo `/var/log/access.log` pode estar vazio devido à rotação do log. Continue seguindo o laboratório como um arquivo `/var/log/access.log` vazio não afetará o laboratório.

- d. Com a **tail** ainda em execução na janela do terminal, clique no ícone do navegador da Web no Dock para abrir uma janela do navegador da Web. Redimensione a janela do navegador da Web de forma a permitir que você veja a parte inferior da janela do terminal onde a **tail** ainda está em execução.

Laboratório - Localização de Arquivos de Log

Nota: Na captura de tela abaixo, a tecla Enter foi pressionada algumas vezes na janela do terminal correndo **tail**. Isto é apenas para visualização, uma vez que a **tail** não processa nenhuma entrada durante a execução com **-f**. As linhas vazias extras facilitam a detecção de novas entradas, pois elas são exibidas na parte inferior da janela do terminal.



- e. Na barra de endereço do navegador da web, digite **127.0.0.1** e pressione Enter. Este é o endereço da própria VM, que informa o navegador para se conectar a um servidor Web em execução no computador local. Uma nova entrada deve ser gravada no arquivo `/var/log/nginx/access.log`. Atualize a página da Web para ver novas entradas adicionadas ao log.

```
127.0.0.1 - - [23/Mar/ 2017:09:48:36 -0400] "GET/HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
```

Como o **tail** ainda está em execução, ele deve exibir a nova entrada na parte inferior da janela do terminal. Além do carimbo de data/hora, sua entrada deve ser parecida com a acima.

Nota: o Firefox armazena páginas em cache para uso futuro. Se uma página já estiver em cache, force o Firefox a ignorar o cache e fazer solicitações da Web, recarregue a página pressionando **<CTRL+SHIFT+R>**.

- f. Como o arquivo de log está sendo atualizado pelo nginx, podemos afirmar com certeza que `/var/log/accs.log` é, de fato, o arquivo de log em uso pelo nginx.
- g. Digite **Ctrl + C** para terminar a sessão de monitoramento de tail.

Etapa 2: BONUS TOOL: Journalctl

A VM CyberOps Workstation é baseada no Arch Linux. Categorizado como uma distribuição Linux, o Arch Linux foi projetado para ser leve, minimalista e simples. Como parte desta filosofia de design, o Arch Linux usa systemd como seu sistema init. No Linux, o processo de inicialização é o primeiro processo carregado quando o computador é inicializado. Init é direta ou indiretamente, o pai de todos os processos em execução no sistema. Ele é iniciado pelo kernel no momento da inicialização e continua a ser executado até que o computador seja desligado. Normalmente, o init tem o ID de processo 1.

Um sistema init é um conjunto de regras e convenções que regem a forma como o espaço do usuário em um determinado sistema Linux é criado e disponibilizado para o usuário. Os sistemas de init também especificam

parâmetros de todo o sistema, como arquivos de configuração global, estrutura de registro e gerenciamento de serviços.

Systemd é um sistema de inicialização moderno projetado para unificar a configuração Linux e o comportamento de serviço em todas as distribuições Linux e tem sido cada vez mais adotado pelas principais distribuições Linux. O Arch Linux depende do systemd para funcionalidade de init. A VM CyberOps Workstation também usa systemd.

system-journald (ou simplesmente journald) é o serviço de log de eventos do systemd e usa somente anexos arquivos binários servindo como seus arquivos de log. Observe que o journald não impede o uso de outros sistemas de registro, como syslog e rsyslog.

Esta seção fornece uma breve visão geral do journalctl, um utilitário journald usado para visualização de logs e monitoramento em tempo real.

- a. Em uma janela de terminal na VM CyberOps Workstation, emita o comando journalctl sem opções para exibir todas as entradas de log do diário (pode ser bastante longo):

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

Dica: No momento, você não está vendo mensagens de outros usuários e do sistema.

Os usuários em grupos 'adm', 'systemd-journal', 'wheel' podem ver todas as mensagens.

Passe -q para desativar este aviso.

```
-- 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.
<alguma saída omitida >
```

A saída começa com uma linha semelhante à abaixo, marcando o timestamp onde o sistema começou a registrar. Observe que os carimbos de data/hora variam de sistema para sistema.

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

O journalctl inclui várias funcionalidades, como rolagem de páginas, mensagens codificadas por cores e muito mais. Use as teclas de seta para cima/para baixo para rolar para cima/para baixo a saída, uma linha de cada vez. Use as teclas de seta do teclado esquerdo/direito para rolar de lado e exibir entradas de log que se estendem além dos limites da janela do terminal. A <ENTER> chave exibe a próxima linha enquanto a barra de espaço exibe a próxima página na saída. Pressione a tecla q para sair do journalctl.

Observe a mensagem de dica fornecida pelo journalctl:

Dica: No momento, você não está vendo mensagens de outros usuários e do sistema.

Os usuários em grupos 'adm', 'systemd-journal', 'wheel' podem ver todas as mensagens.

Passe -q para desativar este aviso.

Laboratório - Localização de Arquivos de Log

Esta mensagem lembra que, como o analista é um usuário regular e não um membro dos grupos `adm`, `systemd-journal` ou `wheel`, nem todas as entradas de log serão exibidas pelo `journalctl`. Ele também afirma que executar `journalctl` com a opção `—q` suprime a mensagem de dica.

Como você pode executar o **journalctl** e ver todas as entradas de log?

- b. **journalctl** inclui opções para ajudar na filtragem da saída. Use a opção `—b` para exibir entradas de log relacionadas à inicialização:

```
[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@andytrtr) (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 0x0000000000100000-0x00000000007ffeffff]
usable
<alguma saída omitida >
```

- c. Para ver entradas relacionadas à última inicialização, adicione o `-1` ao comando acima. Para ver entradas relacionadas às duas últimas inicializações, adicione a opção `-2`.

```
[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 09:35:11 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andytrtr) (gcc
version 6.2.1 20160830 (GCC) ) #1 SMP PREEM
Mar 22 09:35:11 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating
point registers'
Mar 22 09:35:11 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'
```

Laboratório - Localização de Arquivos de Log

```
Mar 22 09:35:11 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
Mar 22 09:35:11 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is
832 bytes, using 'standard' format.
Mar 22 09:35:11 secOps kernel: x86/fpu: Using 'eager' FPU context switches.
Mar 22 09:35:11 secOps kernel: e820: BIOS-provided physical RAM map:
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff]
usable
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff]
reserved
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000ffff]
reserved
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000007ffeff]
usable
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x00000000007fff0000-0x00000000007fffffff]
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
<alguma saída omitida >
```

- d. Use a opção **—list-boots** para listar as inicializações anteriores:

```
[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 48a113d5d2f44979a849c9c0d9ecdaf2 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
<alguma saída omitida >
```

- e. Use **—since “<time range>”** para especificar o intervalo de tempo em que entradas de log devem ser exibidas. Os dois comandos abaixo exibem todas as entradas de log geradas nas últimas duas horas e no último dia, respectivamente:

```
[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. --
```

Laboratório - Localização de Arquivos de Log

```
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 ovssdb-server[263]: 2017-03-31T13:54:45Z|00001|ovssdb_server|INFO|ovssdb-server (Open vSwitch) 2.6.1
Mar 31 09:54:45 secOps ovssdb-server[263]: ovs|00001|ovssdb_server|INFO|ovssdb-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|ovssdb_idl|WARN|Interface table in Open_vSwitch database la
Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-31T13:54:45Z|00006|ovssdb_idl|WARN|Mirror table in Open_vSwitch database lacks
<alguma saída omitida >

[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@andyrttr) (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 Kernel SeCops: x86/fpu: Suporte ao recurso XSAVE 0x002: 'Registros SSE
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 30 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 0x0000000000000000-0x0000000000009fbff] usable
Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
```

Laboratório - Localização de Arquivos de Log

```
Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x000000000000f0000-0x000000000000fffff]
reserved
<alguma saída omitida >
```

- f. **journalctl** também permite exibir entradas de log relacionadas a um serviço específico com a opção **—u**. O comando abaixo exibe entradas de logs relacionadas ao **nginx**:

```
[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.
<alguma saída omitida >
```

Nota: Como parte do **systemd**, os serviços são descritos como unidades. A maioria dos pacotes de instalação de serviço cria unidades e habilita unidades durante o processo de instalação.

- g. Semelhante ao **tail —f**, **journalctl** também suporta monitoramento em tempo real. Use a opção **—f** para instruir **journalctl** a *seguir* um log específico. Pressione **Ctrl+C** para sair.

```
[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
```

Laboratório - Localização de Arquivos de Log

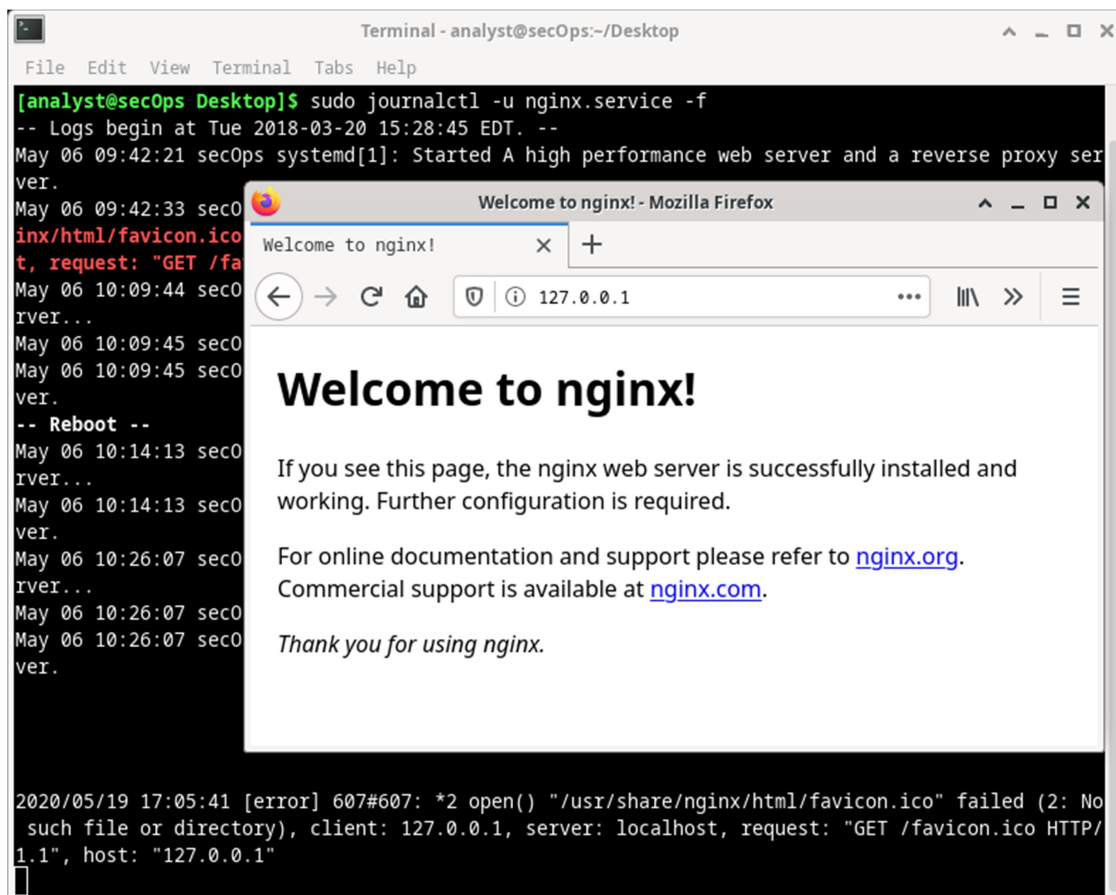
```
Mar 31 10:35:45 secOps filebeat[222]: 2017/03/31 14:35:45.076924 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:36:15 secOps filebeat[222]: 2017/03/31 14:36:15.076060 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:36:45 secOps filebeat[222]: 2017/03/31 14:36:45.076122 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)
<alguma saída omitida >
```

- h. `journalctl` também suporta opções de mistura para obter o conjunto de filtros desejado. O comando abaixo monitora os eventos do sistema `nginx` em tempo real.

```
[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.
```


Laboratório - Localização de Arquivos de Log

- i. Mantenha o comando acima em execução, abra uma nova janela do navegador da Web e digite 127.0.0.1 (configuração padrão) ou 127.0.0.1:8080 (custom_server.conf) na barra de endereços. O journalctl deve exibir um erro relacionado a um arquivo favicon.ico ausente em tempo real. Use Ctrl+C para sair do journalctl.



The screenshot shows a terminal window titled "Terminal - analyst@secOps:~/Desktop" and a Mozilla Firefox browser window titled "Welcome to nginx! - Mozilla Firefox". The terminal displays the output of the command `sudo journalctl -u nginx.service -f`, showing logs for the nginx service. The browser window shows the "Welcome to nginx!" page, which includes instructions on how to use nginx and links to documentation and support. The terminal also shows an error message at the bottom: `2020/05/19 17:05:41 [error] 607#607: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 127.0.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "127.0.0.1"`.

```
Terminal - analyst@secOps:~/Desktop
File Edit View Terminal Tabs Help
[analyst@secOps Desktop]$ sudo journalctl -u nginx.service -f
-- Logs begin at Tue 2018-03-20 15:28:45 EDT. --
May 06 09:42:21 secOps systemd[1]: Started A high performance web server and a reverse proxy server.
May 06 09:42:33 secOps nginx.service: nginx: listening on *
May 06 10:09:44 secOps nginx.service: nginx: http request: GET /favicon.ico
May 06 10:09:45 secOps nginx.service: nginx: http request: GET /favicon.ico
May 06 10:09:45 secOps nginx.service: nginx: http request: GET /favicon.ico
-- Reboot --
May 06 10:14:13 secOps nginx.service: nginx: listening on *
May 06 10:14:13 secOps nginx.service: nginx: http request: GET /favicon.ico
May 06 10:26:07 secOps nginx.service: nginx: http request: GET /favicon.ico
May 06 10:26:07 secOps nginx.service: nginx: http request: GET /favicon.ico
May 06 10:26:07 secOps nginx.service: nginx: http request: GET /favicon.ico
2020/05/19 17:05:41 [error] 607#607: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 127.0.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "127.0.0.1"
```

Reflexão

Arquivos de log são extremamente importantes para solução de problemas.

Localização do arquivo de log segue convenção, mas em última análise, é uma escolha do desenvolvedor.

Na maioria das vezes, as informações do arquivo de log (localização, nomes de arquivo, etc.) estão incluídas na documentação. Se a documentação não fornecer informações úteis sobre arquivos de log, uma combinação de pesquisa na Web e investigação do sistema deve ser usada.

Os relógios devem ser sempre sincronizados para garantir que todos os sistemas tenham a hora correta. Se os relógios não estiverem corretamente definidos, é muito difícil rastrear eventos de volta.

É importante entender quando ocorreram eventos específicos. Além disso, eventos de diferentes fontes são frequentemente analisados ao mesmo tempo.