Travaux pratiques – Consulter les journaux du serveur

# Objectifs

Partie 1: Lire des fichiers journaux à l'aide des programmes Cat, More, Less et Trail

Partie 2 : Fichiers journaux et Syslog

Partie 3 : Fichiers journaux et Journalctl

# Contexte/scénario

Les fichiers journaux sont extrêmement utiles pour dépanner et surveiller les réseaux. Chaque application génère des fichiers journaux différents, avec des champs et des informations qui leur sont propres. Si la structure des champs peut varier selon les fichiers journaux, les outils utilisés pour les lire sont généralement les mêmes. Au cours de ces travaux pratiques, vous allez découvrir les outils couramment utilisés pour lire les fichiers journaux et apprendre à les utiliser.

# Ressources requises

* Poste de travail CyberOps VM

# Instructions

## Lire des fichiers journaux à l'aide des programmes Cat, More, Less et Trail

Les fichiers journaux enregistrent des événements spécifiques déclenchés par les applications, les services ou le système d'exploitation lui-même. Généralement stockés au format de texte clair, les fichiers journaux constituent une ressource indispensable aux opérations de dépannage.

### Ouvrir les fichiers journaux

Les fichiers journaux contiennent généralement des informations en texte clair qui peuvent être affichées par quasiment tous les programmes capables de gérer du texte (des éditeurs de texte, par exemple). Toutefois, pour des raisons de commodité, de simplicité d'utilisation et de vitesse, certains outils sont davantage utilisés que d'autres. Cette section se concentre sur quatre programmes de ligne de commande : **cat**, **more**, **less** et **tail**.

**cat**, dérivé du mot « concaténer », est un outil de ligne de commande UNIX utilisé pour lire et afficher le contenu d'un fichier à l'écran. Le programme **cat** est très largement utilisé à ce jour en raison de sa simplicité d'utilisation et de sa capacité à ouvrir un fichier texte et à l'afficher sur un terminal prenant uniquement en charge du texte.

* + - 1. Démarrez le **poste de travail virtuel CyberOps** et ouvrez une fenêtre du terminal.
      2. Dans la fenêtre du terminal, exécutez la commande suivante pour afficher le contenu du fichier **logstash-tutorial.log** situé dans le dossier **/home/analyst/lab.support.files/** :

analyst@secOps ~$ **cat /home/analyst/lab.support.files/logstash-tutorial.log**

Le contenu du fichier doit défiler dans la fenêtre du terminal, jusqu'à ce que tout le texte soit affiché.

#### Question :

Quel est l'un des inconvénients à utiliser le programme **cat** avec des fichiers texte volumineux ?

Un autre outil populaire d'affichage de fichiers journaux est **more**. Semblable au programme **cat**, **more** est un autre outil de ligne de commande UNIX capable d'ouvrir un fichier texte et d'afficher son contenu à l'écran. La principale différence entre **cat** et **more** réside dans le fait que le programme **more** prend en charge les sauts de page, ce qui permet à l'utilisateur d'afficher le contenu d'un fichier page par page. Il suffit d'appuyer sur la barre d'espace pour afficher la page suivante.

* + - 1. À partir de la même fenêtre du terminal, exécutez la commande suivante pour afficher à nouveau le contenu du fichier **logstash-tutorial.log**. Nous utilisons cette fois-ci la commande **more** :

analyst@secOps ~$ **more /home/analyst/lab.support.files/logstash-tutorial.log**

Le contenu du fichier doit défiler dans la fenêtre du terminal jusqu'à ce que la première page soit affichée. Appuyez sur la barre d'espace pour passer à la page suivante. Appuyez sur Entrée pour afficher la ligne de texte suivante.

#### Question :

Quel est l'inconvénient de la commande **more** ?

Exploitant les fonctionnalités des commandes **cat** et **more**, le programme **less** permet d'afficher le contenu d'un fichier page par page, avec la possibilité de revenir sur les pages précédemment affichées.

* + - 1. À partir de la même fenêtre du terminal, exécutez la commande **less** pour afficher à nouveau le contenu du fichier **logstash-tutorial.log** :

analyst@secOps ~$ **less /home/analyst/lab.support.files/logstash-tutorial.log**

Le contenu du fichier doit défiler dans la fenêtre du terminal jusqu'à ce que la première page soit affichée. Appuyez sur la barre d'espace pour passer à la page suivante. Appuyez sur Entrée pour afficher la ligne de texte suivante. Utilisez les flèches haut et bas pour accéder aux différentes pages du fichier texte.

Utilisez la touche « **q** » de votre clavier pour fermer l'outil **less**.

* + - 1. La commande **tail** affiche la fin d'un fichier texte. Par défaut, la commande **tail** affiche les dix dernières lignes du fichier.

Utilisez la commande « tail » pour afficher les dix dernières lignes du fichier **/home/analyst/lab.support.files/logstash-tutorial.log**.

analyst@secOps ~$ **tail /home/analyst/lab.support.files/logstash-tutorial.log**

218.30.103.62 - - [04/Jan/2015:05:28:43 +0000] "GET /blog/geekery/xvfb-firefox.html HTTP/1.1" 200 10975 "-" "Sogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"

218.30.103.62 - - [04/Jan/2015:05:29:06 +0000] "GET /blog/geekery/puppet-facts-into-mcollective.html HTTP/1.1" 200 9872 "-" "Sogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"

198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/disabling-battery-in-ubuntu-vms.html?utm\_source=feedburner&utm\_medium=feed&utm\_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 9316 "-" "Tiny Tiny RSS/1.11 (http://tt-rss.org/)"

198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/solving-good-or-bad-problems.html?utm\_source=feedburner&utm\_medium=feed&utm\_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 10756 "-" "Tiny Tiny RSS/1.11 (http://tt-rss.org/)"

218.30.103.62 - - [04/Jan/2015:05:29:26 +0000] "GET /blog/geekery/jquery-interface-puffer.html%20target= HTTP/1.1" 200 202 "-" "Sogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"

218.30.103.62 - - [04/Jan/2015:05:29:48 +0000] "GET /blog/geekery/ec2-reserved-vs-ondemand.html HTTP/1.1" 200 11834 "-" "Sogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"

66.249.73.135 - - [04/Jan/2015:05:30:06 +0000] "GET /blog/web/firefox-scrolling-fix.html HTTP/1.1" 200 8956 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 6\_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /projects/xdotool/ HTTP/1.1" 200 12292 "http://www.haskell.org/haskellwiki/Xmonad/Frequently\_asked\_questions" "Mozilla/5.0 (X11; Linux x86\_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"

86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /reset.css HTTP/1.1" 200 1015 "http://www.semicomplete.com/projects/xdotool/" "Mozilla/5.0 (X11; Linux x86\_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"

86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /style2.css HTTP/1.1" 200 4877 "http://www.semicomplete.com/projects/xdotool/" "Mozilla/5.0 (X11; Linux x86\_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"

### Consulter les fichiers journaux de manière dynamique

Dans certains cas, il est utile de consulter les fichiers journaux à mesure que des entrées y sont ajoutées. La commande **tail -f** s'avère alors extrêmement utile.

* + - 1. Utilisez la commande tail -f pour suivre activement le contenu du fichier **/var/log/syslog** :

analyst@secOps ~$ **sudo tail –f /home/analyst/lab.support.files/logstash-tutorial.log**

#### Question :

En quoi les sorties des commandes **tail** et **tail -f** diffèrent-elles ? Expliquez votre réponse.

* + - 1. Pour voir la commande tail -f en action, ouvrez une deuxième fenêtre de terminal. Organisez votre écran de manière à voir les deux fenêtres du terminal. Redimensionnez les fenêtres pour afficher les deux en même temps, comme indiqué dans l'image ci-dessous :

La fenêtre du haut exécute la commande **tail -f** pour surveiller le fichier **/home/analyst/lab.support.files/logstash-tutorial.log**. Utilisez la fenêtre du bas pour ajouter des informations sur le fichier surveillé.

Pour optimiser l'affichage du contenu, sélectionnez la fenêtre du haut (celle qui exécute la commande **tail -f**) et appuyez plusieurs fois sur Entrée. Cela insérera quelques lignes entre le contenu actuel du fichier et les nouvelles informations à ajouter.

* + - 1. Sélectionnez la fenêtre de terminal du bas et saisissez la commande suivante :

[analyst@secOps ~]$ **echo "ceci est une nouvelle entrée du fichier journal surveillé" >> lab.support.files/logstash-tutorial.log**

La commande ci-dessus ajoute le message « ceci est une nouvelle entrée du fichier journal surveillé » au fichier **/home/analyst/lab.support.files/logstash-tutorial.log**. Comme la commande tail -f surveille actuellement le fichier, une ligne est ajoutée au fichier. La fenêtre du haut devrait afficher la nouvelle ligne en temps réel.

* + - 1. Appuyez sur CTRL+C pour arrêter l'exécution de la commande **tail -f** et revenir à l'invite shell.
      2. Fermez l'une des deux fenêtres du terminal.

## Fichiers journaux et Syslog

En raison de leur grande utilité, les fichiers journaux sont généralement stockés sur le même ordinateur de surveillance. **Syslog** est un système conçu pour permettre aux périphériques d'envoyer leurs fichiers journaux sur un serveur centralisé, appelé serveur **syslog**. Les clients communiquent avec le serveur syslog via le protocole **syslog**. **Syslog** est largement déployé dans les entreprises et prend en charge la quasi-totalité des plates-formes informatiques.

Le poste de travail virtuel CyberOps génère des fichiers journaux associés au système d'exploitation et les transmet au serveur **syslog**.

* + - 1. Utilisez la commande **cat** en tant que commande **root** pour répertorier le contenu du fichier **/var/log/syslog**. Ce fichier contient les entrées de journal qui sont générées par le système d'exploitation du poste de travail virtuel CyberOps et transmises au service **syslog**.

analyst@secOps ~$ **sudo cat /var/log/syslog.1**

[sudo] password for analyst:

Feb 7 13:23:15 secOps kernel: [ 5.458959] psmouse serio1: hgpk: ID: 10 00 64

Feb 7 13:23:15 secOps kernel: [ 5.467285] input: ImExPS/2 BYD TouchPad as /devices/platform/i8042/serio1/input/input6

Feb 7 13:23:15 secOps kernel: [ 5.502469] RAPL PMU: API unit is 2^-32 Joules, 4 fixed counters, 10737418240 ms ovfl timer

Feb 7 13:23:15 secOps kernel: [ 5.502476] RAPL PMU: hw unit of domain pp0-core 2^-0 Joules

Feb 7 13:23:15 secOps kernel: [ 5.502478] RAPL PMU: hw unit of domain package 2^-0 Joules

Feb 7 13:23:15 secOps kernel: [ 5.502479] RAPL PMU: hw unit of domain dram 2^-0 Joules

Feb 7 13:23:15 secOps kernel: [ 5.502480] RAPL PMU: hw unit of domain pp1-gpu 2^-0 Joules

Feb 7 13:23:15 secOps kernel: [ 5.672547] ppdev: user-space parallel port driver

Feb 7 13:23:15 secOps kernel: [ 5.709000] pcnet32 0000:00:03.0 enp0s3: renamed from eth0

Feb 7 13:23:16 secOps kernel: [ 6.166738] pcnet32 0000:00:03.0 enp0s3: link up, 100Mbps, full-duplex

Feb 7 13:23:16 secOps kernel: [ 6.706058] random: crng init done

Feb 7 13:23:18 secOps kernel: [ 8.318984] floppy0: no floppy controllers found

Feb 7 13:23:18 secOps kernel: [ 8.319028] work still pending

Feb 7 14:26:35 secOps kernel: [ 3806.118242] hrtimer: interrupt took 4085149 ns

Feb 7 15:02:13 secOps kernel: [ 5943.582952] pcnet32 0000:00:03.0 enp0s3: link down

Feb 7 15:02:19 secOps kernel: [ 5949.556153] pcnet32 0000:00:03.0 enp0s3: link up, 100Mbps, full-duplex

#### Question :

Pourquoi fallait-il exécuter la commande **cat** en tant que commande **root** ?

* + - 1. Vous remarquerez que le fichier **/var/log/syslog** stocke uniquement les entrées de journal les plus récentes. Pour limiter la taille du fichier syslog, le système d'exploitation effectue une rotation des fichiers journaux. Pour cela, il renomme les fichiers journaux les plus anciens **syslog.1**, **syslog.2** et ainsi de suite.

Utilisez la commande **cat** pour afficher les fichiers **syslog** les moins récents :

analyst@secOps ~$ **sudo cat /var/log/syslog.2**

analyst@secOps ~$ **sudo cat /var/log/syslog.3**

analyst@secOps ~$ **sudo cat /var/log/syslog.4**

#### Question :

D'après vous, pourquoi est-il si important de toujours synchroniser l'heure et la date des ordinateurs ?

## Fichiers journaux et Journalctl

**journal** est un autre système de gestion des événements couramment utilisé. Géré par le démon **journald**, le système est conçu pour centraliser la gestion des événements, peu importe d'où proviennent les messages. Dans le cadre de ce TP, la fonctionnalité du démon système **journal** qui nous intéresse le plus est l'utilisation de fichiers binaires append-only en tant que **fichiers journaux**.

### Exécuter journalctl sans aucune option

* + - 1. Pour consulter les journaux **journald**, utilisez la commande **journalctl**. L'outil **journalctl** interprète et affiche les entrées de journal stockées dans les fichiers journaux binaires **journal**.

analyst@secOps ~$ **journalctl**

-- Logs begin at Fri 2014-09-26 14:13:12 EDT, end at Tue 2017-02-07 13:23:29 ES

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.

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

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

Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopped target Basic System.

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

Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopped target Paths.

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

Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopped target Timers.

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

<output omitted>

**Remarque** : le fait d'exécuter la commande journalctl en tant que commande root permet d'afficher des informations plus détaillées.

* + - 1. Appuyez sur CTRL+C pour quitter l'écran.

### Journalctl et ses options

La commande **journalctl** fournit un certain nombre d'options puissantes. Pour les commandes suivantes, utilisez CRTL+C pour quitter l'écran.

* + - 1. Utilisez la commande **journalctl -utc** pour afficher tous les horodatages au format UTC:

analyst@secOps ~$ **sudo journalctl -–utc**

* + - 1. Utilisez la commande **journalctl -b** pour afficher les entrées de journal enregistrées lors du dernier démarrage du système :

analyst@secOps ~$ **sudo journalctl –b**

Feb 07 08:23:13 secOps systemd-journald[172]: Time spent on flushing to /var is

Feb 07 08:23:13 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andyrtr)

Feb 07 08:23:13 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 fl

Feb 07 08:23:13 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE re

Feb 07 08:23:13 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX re

Feb 07 08:23:13 secOps kernel: x86/fpu: xstate\_offset[2]: 576, xstate\_sizes[2]

Feb 07 08:23:13 secOps kernel: x86/fpu: Enabled xstate features 0x7, context si

Feb 07 08:23:13 secOps kernel: x86/fpu: Using 'eager' FPU context switches.

Feb 07 08:23:13 secOps kernel: e820: BIOS-provided physical RAM map:

<output omitted>

* + - 1. Utilisez la commande **journalctl** pour définir le service et la période de consignation des entrées de journal. La commande suivante affiche tous les journaux de service **nginx** enregistrés aujourd'hui :

analyst@secOps ~$ **sudo journalctl -u nginx.service --since today**

* + - 1. Utilisez l'option **-k** pour afficher uniquement les messages générés par le noyau :

analyst@secOps ~$ **sudo journalctl –k**

* + - 1. Comme dans la commande **tail -f** décrite précédemment, utilisez l'option **-f** pour surveiller activement les journaux à mesure que de nouvelles entrées y sont écrites:

analyst@secOps ~$ **sudo journalctl –f**

# Question de réflexion

Comparez les outils Syslog et Journald. Quels sont les avantages et les inconvénients de chaque outil ?

Fin du document