CHALLENGE HONEYNET 5 : ANALYSE DE LOGS

GNU/Linux Magazine n° 139 | juin 2011 | Christophe Grenier

Fin 2010, le projet Honeynet a proposé son 5ème challenge de l'année : une analyse de logs. Cette épreuve a été annoncée comme nécessitant un niveau intermédiaire de connaissance. En pratique, aucun participant n'a donné de résultats satisfaisants ! Analyser correctement des logs est plus difficile que l'on ne le croit et c'est pour cela que je vous propose de découvrir en détail ce challenge.

1. PREMIÈRE ANALYSE

Avoir avoir récupéré et décompressé l'archive (**http://www.honeynet.org/files/sanitized_log.zip**) contenant les logs, nous obtenons les fichiers suivants :

```
udev
debug
dpkg.log
user.log
messages
apt/term.log
secure
dmesg.0
auth.log
kern.log
fsck/checkfs
fsck/checkroot
fontconfig.log
dmesg
apache2/www-media.log
apache2/www-error.log
apache2/www-access.log
daemon.log
```

Les noms de fichiers sont révélateurs :

- **dmesg**, **fsck**, **secure**... indiquent qu'il s'agit d'un système Unix ;

- **dpkg** et **apt** sont utilisés par les distributions Linux de type Debian et Ubuntu ;

- présence du serveur web apache2.

Pour chaque fichier, commençons par être attentif aux dates des enregistrements.

1.1 UDEV

```
export TZ=UTC
[kmaster@adsl sanitized_log]$ grep "UEVENT\[" udev |head -1
UEVENT[1272866735.318771] add      /bus/acpi (bus)
[kmaster@adsl sanitized_log]$ grep "UEVENT\[" udev |tail -1
UEVENT[1272866738.063339] add      /bus/pci/drivers/parport_pc (drivers)
```

Le moment des événements est exprimé en secondes depuis Epoch, le 1er janvier 1970. Convertissons les dates.

```
[kmaster@adsl sanitized_log]$ date -u -d "1970-01-01 1272866738.063339 secs"
Mon May 3 06:05:38 UTC 2010
[kmaster@adsl sanitized log]$ date -u -d "1970-01-01 1272866738.063339 secs"
Mon May 3 06:05:38 UTC 2010
[kmaster@adsl sanitized_log]$ ls -l udev
-rw-r--r--. 1 kmaster kmaster 359696 May 3 2010 udev
```

1.2 DEBUG

Ce fichier ne présente pas d'intérêt. Toutes les informations qu'il contient en dehors de ces 5 lignes sont présentes dans le fichier **kern.log**.

```
<pre>
[kmaster@adsl sanitized log]$ diff -uw debug kern.log |grep ^-
--- debug       2010-07-03 19:59:31.000000000 +0200
-Mar 22 13:49:50 app-1 ntpd[5894]: signal_no_reset: signal 17 had flags 4000000
-Mar 22 13:51:05 app-1 ntpd[6073]: signal no reset: signal 17 had flags 4000000
-Mar 22 18:43:42 app-1 ntpd[5037]: signal no reset: signal 17 had flags 4000000
-Mar 22 18:45:26 app-1 ntpd[5038]: signal no reset: signal 17 had flags 4000000
-Apr 18 18:04:00 app-1 ntpd[4929]: signal_no_reset: signal 17 had flags 4000000
</pre>
```

1.3 DPKG.LOG ET APT/TERM.LOG

Ces deux fichiers listent des installations de packages pour Ubuntu entre **2010-04-19 12:00:17** et **2010-04-26 04:53:23**.

1.4 USER.LOG

Ce fichier ne comporte que trois lignes, la zone horaire est America/Los_Angeles.

1.5 MESSAGES

En dehors des lignes suivantes, tous les enregistrements sont présents dans le fichier **kern.log**.

```
[kmaster@adsl sanitized log]$ diff -uw messages kern.log |grep ^-
--- messages    2010-05-03 08:07:22.000000000 +0200
-Apr 28 07:34:22 app-1 syslog-ng[4566]: syslog-ng starting up; version='3.0.5'
-Apr 28 09:35:32 app-1 syslog-ng[4566]: Termination requested via signal, terminating;
-Apr 28 09:35:32 app-1 syslog-ng[4566]: syslog-ng shutting down; version='3.0.5'
-May 2 23:05:47 app-1 syslog-ng[4547]: syslog-ng starting up; version='3.0.5'
```

1.6 SECURE

Ce fichier porte la date du 25 avril, mais problème, il est vide !

1.7 DMESG ET DMESG.0

Ces informations sur le démarrage du système se retrouvent dans le fichier **kern.log**.

1.8 AUTH.LOG, KERN.LOG

Des informations du 16 mars au 2 mai se trouvent dans ces fichiers, mais ils ont été modifiés le 3 juillet pour la dernière fois.

1.9 DAEMON.LOG

Des informations du 16 mars au 2 mai se trouvent dans ce fichier. Cependant, lui aussi comporte une date de modification du 3 juillet.

1.10 FSCK/CHECKFS ET FSCK/CHECKROOT

Un **fsck** a été réalisé le 2 mai.

1.11 FONTCONFIG.LOG

Rien d'intéressant.

1.12 APACHE2/WWW-MEDIA.LOG, APACHE2/WWW-ERROR.LOG ET APACHE2/WWW-ACCESS.LOG

Ces fichiers comportent des enregistrements du 19 au 24 avril. Mais là encore, ces fichiers ont été modifiés le 3 juillet.

1.13 EN RÉSUMÉ

Lorsque l'on doit analyser des logs, on doit se demander si :

- des enregistrements ont été supprimés, altérés ou ajoutés ;

- l'horodatage est fiable.

Plusieurs fichiers comportent une date postérieure à celle des derniers enregistrements. Comme le répertoire contenant ces fichiers s'appelle **`sanitized_log`**, on comprend que les organisateurs du challenge aient anonymisé certaines informations.

Tout ces logs sont a priori issus de la même machine et les fichiers **`daemon.log`** et **`debug`** indiquent que le démon de synchronisation horaire **`ntpd`** est utilisé. Le fuseau horaire est celui de Los_Angeles, nous pouvons donc supposer que l'horodatage est fiable, mais attention, cela n'est pas toujours le cas :

```
Mar 23 23:19:29 app-1 ntpd[5285]: time correction of 22587 seconds exceeds sanity limit
(1000); set clock manually to the correct UTC time.
Mar 24 16:48:49 app-1 ntpdate[16526]: step time server 91.189.94.4 offset 22586.919842 sec
```

La première question que l'on nous pose est de savoir si le système a été compromis et quand.

Il peut s'agir d'une compromission antérieure au début des logs, visible ou non dans ceux-ci, ou d'attaques s'étant produites sur la période couverte par ces enregistrements.

Un système peut être compromis par :

- des attaques locales hors ligne (par exemple, brancher le disque dur sur un système sous son contrôle) ;

- des attaques locales en ligne (escalade de privilèges d'un utilisateur authentifié) ;

- des attaques réseau ;

- des attaques d'un applicatif utilisateur (exploitation d'une vulnérabilité du navigateur web, …).

Nous avons constaté que le fichier **`secure`** était vide, nous savons donc que le serveur a été compromis.

2. KERN.LOG: VM, REDÉMARRAGE DU SERVEUR ET BACKDOOR

Ce fichier de log provient d'une machine virtuelle VMWare :

```
Mar 16 08:09:58 app-1 kernel: [    0.000000] ACPI: SRAT 1FEF0909, 0080 (r2 VMWARE MEMPLUG
  6040000 VMW         1)
```

Le fichier **`kern.log`** nous permet d'apprendre à quel moment le serveur démarre :

```
[kmaster@adsl sanitized_log]$ grep Inspecting kern.log
Mar 16 08:09:58 app-1 kernel: Inspecting /boot/System.map-2.6.24-26-server
Mar 18 09:41:44 app-1 kernel: Inspecting /boot/System.map-2.6.24-26-server
Mar 18 09:48:54 app-1 kernel: Inspecting /boot/System.map-2.6.24-26-server
Mar 18 09:50:23 app-1 kernel: Inspecting /boot/System.map-2.6.24-26-server
```

```
Mar 18 09:54:25 app-1 kernel: Inspecting /boot/System.map-2.6.24-26-server
```

Des erreurs de segmentation (segfault) pour les démons **sshd** et **collectd** (collecte de statistiques systèmes) sont listées :

```
[kmaster@adsl sanitized_log]$ grep -i segfault kern.log
Apr 21 12:12:24 app-1 kernel: : [237397.126529] sshd[2798]: segfault at 0 rip 8048e33 rsp
ffdf4600 error 4
...
Apr 28 09:35:31 app-1 kernel: : [ 7282.658888] collectd[5120]: segfault at 0 rip 40a0d9 rsp
7fff68dd3790 error 4
```

On peut en déduire qu'au 21 avril, le démon **sshd** a été remplacé par une version fournie par un pirate. A priori, cet exécutable n'est pas compatible ou sérieusement bogué.

En général, le binaire est souvent remplacé pour capturer les mots de passe utilisateurs et ouvrir un accès root avec un couple compte/mot de passe ne dépendant pas des fichiers **/etc/passwd** et **/etc/shadow**.

3. MANIPULATION DE COMPTES

Recherchons les programmes ayant produit des logs enregistrés dans **auth.log** :

```
[kmaster@adsl sanitized log]$ awk '{ sub(/\[.+/,"",$5); print $5}' auth.log|sort -u
chage
chfn
chsh
CRON
groupadd
login
passwd
sshd
su
sudo:
useradd
userdel
usermod
```

Concentrons-nous pour le moment sur les créations, suppressions de compte, changements de mot de passe, ...

Le compte **user4** est supprimé aussitôt créé :

```
[kmaster@adsl sanitized_log]$ egrep -e
'(chage|chfn|chsh|groupadd|passwd|useradd|userdel|usermod)\[' auth.log
Mar 16 08:12:13 app-1 groupadd[4691]: new group: name=user4, GID=1001
Mar 16 08:12:13 app-1 useradd[4692]: new user: name=user4, UID=1001, GID=1001,
home=/home/user4, shell=/bin/bash
Mar 16 08:12:17 app-1 passwd[4695]: pam_unix(passwd:chauthtok): password changed for user4
Mar 16 08:12:22 app-1 chfn[4696]: changed user `user4' information
Mar 16 08:12:31 app-1 userdel[4700]: delete user `user4'
Mar 16 08:12:31 app-1 userdel[4700]: removed group `user4' owned by `user4'
```

Création des comptes **user1** et **user2** :

```
Mar 16 08:12:38 app-1 groupadd[4702]: new group: name=user1, GID=1001
Mar 16 08:12:38 app-1 useradd[4703]: new user: name=user1, UID=1001, GID=1001,
home=/home/user1, shell=/bin/bash
Mar 16 08:12:44 app-1 passwd[4706]: pam unix(passwd:chauthtok): password changed for user1
Mar 16 08:12:46 app-1 chfn[4707]: changed user `user1' information
Mar 16 08:12:49 app-1 chfn[4708]: changed user `user1' information
Mar 16 08:12:55 app-1 groupadd[4710]: new group: name=user2, GID=1002
Mar 16 08:12:55 app-1 useradd[4711]: new user: name=user2, UID=1002, GID=1002,
home=/home/user2, shell=/bin/bash
Mar 16 08:13:00 app-1 passwd[4714]: pam unix(passwd:chauthtok): password changed for user2
Mar 16 08:13:02 app-1 chfn[4715]: changed user `user2' information
```

Nous allons maintenant vivre à travers ces logs les premiers moments de l'installation de cette machine.

Installation de **sshd** :

```
Mar 16 08:25:22 app-1 useradd[4845]: new user: name=sshd, UID=104, GID=65534,
home=/var/run/sshd, shell=/usr/sbin/nologin
Mar 16 08:25:22 app-1 usermod[4846]: change user `sshd' password
Mar 16 08:25:22 app-1 chage[4847]: changed password expiry for sshd
```

Modification du mot de passe de **user1** :

```
Mar 18 10:00:06 app-1 passwd[4763]: pam_unix(passwd:chauthtok): password changed for user1
```

Installation d'une messagerie exim :

```
Mar 18 10:15:42 app-1 groupadd[5392]: new group: name=Debian-exim, GID=114
Mar 18 10:15:42 app-1 useradd[5393]: new user: name=Debian-exim, UID=105, GID=114,
home=/var/spool/exim4, shell=/bin/false
Mar 18 10:15:43 app-1 chage[5394]: changed password expiry for Debian-exim
```

- Installation d'Apache :

```
Mar 18 10:17:01 app-1 chsh[6866]: changed user `www-data' shell to `/bin/bash'
```

- Installation de MySQL :

```
Mar 18 10:18:26 app-1 groupadd[6963]: new group: name=mysql, GID=115
Mar 18 10:18:26 app-1 useradd[6966]: new user: name=mysql, UID=106, GID=115,
home=/var/lib/mysql, shell=/bin/false
Mar 18 10:18:26 app-1 chage[6967]: changed password expiry for mysql
Mar 18 10:18:26 app-1 chfn[6968]: changed user `mysql' information
```

Changement du mot de passe de **user2** :

```
Mar 18 11:39:40 app-1 passwd[10178]: pam_unix(passwd:chauthtok): password changed for user2
```

Changement du mot de passe du compte **root** :

```
Mar 29 13:27:22 app-1 passwd[21555]: pam_unix(passwd:chauthtok): password changed for root
Apr 19 11:04:01 app-1 passwd[30283]: pam_unix(passwd:chauthtok): password changed for root
```

Nous avons effectué un saut de 3 semaines entre ces deux enregistrements, et nous le verrons, ces trois semaines n'ont pas été calmes.

Création d'un compte avec les droits **root** (**uid=0**, **gid=0**) nommé **packet**, un pirate à l'œuvre !

```
Apr 19 22:38:00 app-1 useradd[2019]: new user: name=packet, UID=0, GID=0, home=/home/packet,
shell=/bin/sh
Apr 19 22:38:09 app-1 passwd[2020]: pam_unix(passwd:chauthtok): password changed for packet
Apr 19 22:38:50 app-1 passwd[2024]: pam_unix(passwd:chauthtok): new password not acceptable
```

Création du compte **dhg** :

```
Apr 19 22:45:13 app-1 groupadd[2052]: new group: name=dhg, GID=1003
Apr 19 22:45:13 app-1 useradd[2053]: new user: name=dhg, UID=1003, GID=1003, home=/home/dhg,
shell=/bin/bash
Apr 19 22:45:23 app-1 passwd[2056]: pam_unix(passwd:chauthtok): password changed for dhg
Apr 19 22:45:47 app-1 chfn[2057]: changed user `dhg' information
```

Installation de **dbus** :

```
Apr 24 19:27:34 app-1 groupadd[1385]: new group: name=messagebus, GID=117
Apr 24 19:27:35 app-1 useradd[1386]: new user: name=messagebus, UID=108, GID=117,
home=/var/run/dbus, shell=/bin/false
Apr 24 19:27:35 app-1 usermod[1387]: change user `messagebus' password
Apr 24 19:27:35 app-1 chage[1388]: changed password expiry for messagebus
```

Création du compte **fido** :

```
Apr 25 10:41:44 app-1 useradd[9596]: new group: name=fido, GID=1004
Apr 25 10:41:44 app-1 useradd[9596]: new user: name=fido, UID=0, GID=1004, home=/home/fido,
shell=/bin/sh
Apr 25 10:43:24 app-1 passwd[9864]: pam_unix(passwd:chauthtok): password changed for fido
```

Peu après la connexion de 188.131.23.37 sur le compte **root**, création du compte **wind3str0y** :

```
Apr 26 04:43:15 app-1 groupadd[20114]: new group: name=wind3str0y, GID=1005
Apr 26 04:43:15 app-1 useradd[20115]: new user: name=wind3str0y, UID=1004, GID=1005,
home=/home/wind3str0y, shell=/bin/bash
Apr 26 04:43:31 app-1 passwd[20119]: pam_unix(passwd:chauthtok): password changed for
wind3str0y
Apr 26 04:43:34 app-1 chfn[20120]: changed user `wind3str0y' information
```

Revenons sur l'installation de **dbus**. Le fichier **apt/term.log** indique l'installation de nombreux packages dans celui-ci :

```
Log started: 2010-04-24 19:26:39
...
Unpacking dbus (from .../dbus_1.1.20-1ubuntu3.3_amd64.deb) ...
...
Setting up dbus (1.1.20-1ubuntu3.3) ...
Adding system user `messagebus' (UID 108) ...
Adding new group `messagebus' (GID 117) ...
Adding new user `messagebus' (UID 108) with group `messagebus' ...
Not creating home directory `/var/run/dbus'.
```

4. MISE EN PLACE DU SERVEUR

Observons les premières connexions réseau. Deux utilisateurs sont connectés simultanément :

- **user1**, console **pts/0**, depuis 76.191.195.140, n'ayant pas le droit d'utiliser la commande **sudo**, **user NOT in sudoers**.

- **user3**, console **pts/1**, depuis 10.0.1.2, connaissant le mot de passe **root**.

```
Mar 18 10:00:10 app-1 sshd[4764]: Accepted password for user1 from 76.191.195.140 port 35226
ssh2
Mar 18 10:00:10 app-1 sshd[4766]: pam unix(sshd:session): session opened for user user1 by
(uid=0)
Mar 18 10:00:30 app-1 sshd[4786]: Accepted password for user3 from 10.0.1.2 port 64950 ssh2
Mar 18 10:00:30 app-1 sshd[4788]: pam_unix(sshd:session): session opened for user user3 by
(uid=0)
Mar 18 10:00:36 app-1 sudo:     user3 : TTY=pts/1 ; PWD=/home/user3 ; USER=root ;
COMMAND=/bin/su
Mar 18 10:00:36 app-1 sudo: pam_unix(sudo:session): session opened for user root by
user3(uid=0)
Mar 18 10:00:36 app-1 sudo: pam unix(sudo:session): session closed for user root
Mar 18 10:00:36 app-1 su[4805]: Successful su for root by root
Mar 18 10:00:36 app-1 su[4805]: + pts/1 root:root
Mar 18 10:00:36 app-1 su[4805]: pam unix(su:session): session opened for user root by
user3(uid=0)
Mar 18 10:01:03 app-1 sudo:   user1 : user NOT in sudoers ; TTY=pts/0 ; PWD=/home/user1 ;
USER=root ; COMMAND=/bin/su -
```

Le fichier **wtmp** manque à l'appel, il aurait été pratique de l'avoir sous la main pour observer les connexions et deconnexions des utilisateurs. Les traces que nous observons sont moins faciles à suivre.

Une minute plus tard, l'utilisateur **user1** a le droit de se connecter ! Comme l'adresse 10.0.1.2 est privée, nous pouvons supposer qu'elle appartient à un utilisateur légitime. **user3** a réalisé de la configuration pour l'utilisateur **user1**, 76.191.195.140 est donc une adresse IP d'un utilisateur légitime.

```
Mar 18 10:02:09 app-1 sudo:   user1 : TTY=pts/0 ; PWD=/home/user1 ; USER=root ;
COMMAND=/bin/su -
Mar 18 10:02:09 app-1 sudo: pam_unix(sudo:session): session opened for user root by
user1(uid=0)
```

```
Mar 18 10:02:09 app-1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 10:02:09 app-1 su[4875]: Successful su for root by root
Mar 18 10:02:09 app-1 su[4875]: + pts/0 root:root
Mar 18 10:02:09 app-1 su[4875]: pam_unix(su:session): session opened for user root by
user1(uid=0)
```

Différents logiciels sont installés, dont le système de messagerie exim et la base de données MySQL :

```
Mar 18 10:15:42 app-1 groupadd[5392]: new group: name=Debian-exim, GID=114
Mar 18 10:15:42 app-1 useradd[5393]: new user: name=Debian-exim, UID=105, GID=114,
home=/var/spool/exim4, shell=/bin/false
Mar 18 10:15:43 app-1 chage[5394]: changed password expiry for Debian-exim
Mar 18 10:17:01 app-1 chsh[6866]: changed user `www-data' shell to `/bin/bash'
Mar 18 10:18:26 app-1 groupadd[6963]: new group: name=mysql, GID=115
Mar 18 10:18:26 app-1 useradd[6966]: new user: name=mysql, UID=106, GID=115,
home=/var/lib/mysql, shell=/bin/false
Mar 18 10:18:26 app-1 chage[6967]: changed password expiry for mysql
Mar 18 10:18:26 app-1 chfn[6968]: changed user `mysql' information
Mar 18 10:32:44 app-1 su[4875]: pam_unix(su:session): session closed for user root
Mar 18 10:32:49 app-1 su[4805]: pam_unix(su:session): session closed for user root
```

Le gestionnaire de sources SVN est utilisé pour gérer une application web :

```
Mar 18 10:41:13 app-1 sudo:   user1 : TTY=pts/0 ; PWD=/opt/software/web/app ; USER=root ;
COMMAND=/usr/bin/svn commit -m settingsdebug should only change the necessary fields, not
redefine entire LOGGING statement --username user4
```

Une tentative de connexion sur l'utilisateur existant **`user2`** a lieu depuis 71.132.129.212, mais le mot de passe fourni ne passe pas :

```
Mar 18 11:38:04 app-1 sshd[10156]: pam_unix(sshd:auth): authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=adsl-71-132-129-212.dsl.pltn13.pacbell.net user=user2
Mar 18 11:38:05 app-1 sshd[10156]: Failed password for user2 from 71.132.129.212 port 34624
ssh2
```

Le mot de passe du compte **`user2`** est changé, 71.132.129.212 peut se connecter:

```
Mar 18 11:39:40 app-1 passwd[10178]: pam_unix(passwd:chauthtok): password changed for user2
Mar 18 11:39:50 app-1 sshd[10158]: Accepted password for user2 from 71.132.129.212 port 34333
ssh2
```

Sur le serveur, nous avons donc :

- **`tty1 user3 -> root`** ;
- **`pts/0 76.191.195.140 user1`** ;
- **`pts/1 10.0.1.2 user3 -> root`** ;
- **`pts/2 71.132.129.212 user2`**.

La coopération entre l'utilisateur se trouvant sur la console et certains se connectant en SSH nous permettra de considérer les utilisateurs venant d'adresse IP privée comme légitime ainsi que ceux se connectant sur les comptes **`user1`**, **`user2`** et **`user3`**. Ces noms de comptes ont été anonymisés. Nous pouvons donc considérer que ceux qui ne le sont pas, en dehors de **`root`**, ne sont pas légitimes.

5. SSH: FRIEND OR FOE ?

Récupérons toutes les adresses ayant tenté, avec succès ou non, de se connecter au serveur SSH. Nous obtenons 60 adresses différentes.

Nous distinguons dans les logs SSH les connexions valides, les échecs d'authentification liés à un mauvais mot de passe et ceux liés à un nom d'utilisateur inexistant :

```
Mar 16 08:26:06 app-1 sshd[4894]: Accepted password for user3 from 192.168.126.1 port 61474
ssh2
Mar 18 11:38:05 app-1 sshd[10156]: Failed password for user2 from 71.132.129.212 port 34624
ssh2
Apr 19 04:36:51 app-1 sshd[6990]: Failed password for invalid user tomcat from 203.81.226.86
port 59207 ssh2
```

À l'aide de ces informations, essayons de classer ces adresses en :

- utilisateurs légitimes ;

- suspects, ceux que nous n'avons pas réussi à classer ;

- intrus.

En conservant cet ordre, nous partons donc avec le score 0/60/0.

- Identifions chaque adresse IP ayant tenté de se connecter mais n'ayant jamais réussi à s'authentifier comme des pirates. Score : 0 29 31.

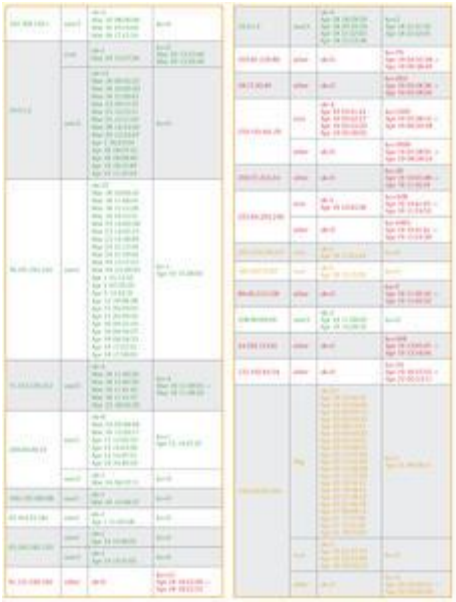
- Plus d'échecs de connexion que de réussites et au moins 5 échecs. Score : 0/23/37.

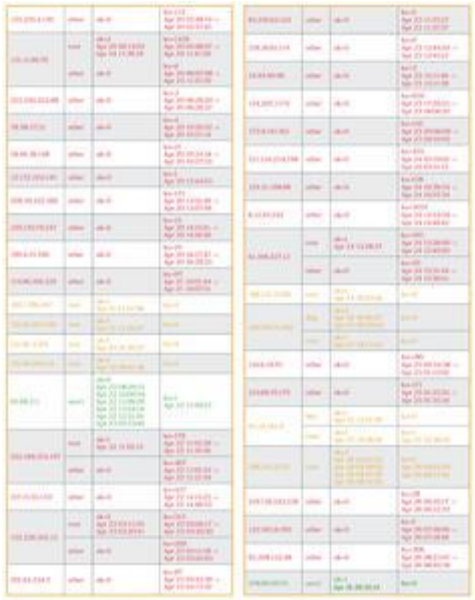
En ne considérant que les adresses IP ayant eu au moins 5 échecs d'authentification, nous évitons de classer comme pirate un utilisateur ayant des problèmes pour taper son mot de passe. Cela nous permet d'identifier des adresses ayant réalisé des attaques par force brute.

- Les adresses IP privées encore non classées sont des gentils : 3/20/37.

- Utilise les comptes **`user1`**, **`user2`** ou **`user3`**. Le score devient 12/11/37.

Présentons les résultats :





Il reste donc 11 suspects :

- 201.229.176.217 - Apr 19 11:03:44

```
Apr 19 11:04:01 app-1 passwd[30283]: pam_unix(passwd:chauthtok): password changed for root
```

Cette personne a changé le mot de passe, cependant nous ne savons pas s'il s'agit d'un organisateur du challenge qui aurait modifié le mot de passe directement dans le fichier **`shadow`** ou d'un pirate voulant bloquer l'accès à l'administrateur ou à d'autres pirates.

- 190.167.70.87 - Apr 19 11:15:26

- 190.166.87.164 - Apr 19 22:37:24, crée le compte **`dhg`** et se connecte dessus, pirate !

- 193.1.186.197 - Apr 21 11:51:38

- 151.81.205.100 - Apr 21 11:56:37

- 151.82.3.201 - Apr 22 01:30:27

- 151.81.204.141 - Apr 22 06:41:38

- 188.131.22.69 - Apr 24 16:33:36

- 190.167.74.184 - Apr 24 18:46:57 se connecte sur le compte **`dhg`**

- 94.52.185.9 - Apr 25 10:38:56, crée le compte **`fido`** et se connecte dessus, pirate !

- 188.131.23.37 - Apr 26 04:42:50, crée le compte **`wind3str0y`**, pirate !

Parmi ces 11 suspects, nous venons d'en identifier quatre comme des pirates. Le score devient 12/7/41. Je serais bien tenté de classer ces 7 adresses IP restantes ayant toutes réussi à se connecter sur le compte **`root`** comme des pirates, car elles ne dépendent pas des providers des utilisateurs légitimes actuellement identifiés, mais en toute honnêteté, je n'ai pas d'éléments concrets.

6. APACHE

Établissons la liste des URL enregistrées dans **`apache2/www-access.log`** :

```
[kmaster@adsl sanitized_log]$ cut -d\" -f2 apache2/www-access.log|sort -u
CONNECT 72.51.18.254:6677 HTTP/1.0
CONNECT 92.62.43.77:6667 HTTP/1.0
GET /feed/ HTTP/1.1
```

```
GET / HTTP/1.1
GET http://72.51.18.254:6677 HTTP/1.0
GET http://edit.yahoo.com/config?login=DoomsDay&passwd=youcanrap HTTP/1.0
GET http://proxyjudge1.proxyfire.net/fastenv HTTP/1.1
GET http://www.wantsfly.com/prx2.php?hash=FABB83E72D135F1018046CC4005088B36F8D0BEDCEA7
HTTP/1.0
GET http://www.wantsfly.com/prx2.php HTTP/1.0
GET /login/ HTTP/1.1
GET /robots.txt HTTP/1.1
GET /signup/ HTTP/1.1
GET /wp-admin HTTP/1.1
GET /wp-admin/ HTTP/1.1
POST /signup/ HTTP/1.1
POST /wp-cron.php?doing_wp_cron HTTP/1.0
```

Nous pouvons distinguer l'utilisation de WordPress et des tentatives d'utiliser le serveur web comme proxy HTTP.

```
193.109.122.56 - - [20/Apr/2010:00:00:01 -0700] "CONNECT 72.51.18.254:6677 HTTP/1.0" 301 - "-"
"pxyscand/2.1" oFs91QoAAQ4AAAQFlmcAAAAL 1213441
193.109.122.33 - - [20/Apr/2010:00:00:05 -0700] "GET http://72.51.18.254:6677 HTTP/1.0" 400
401 "-" "-" - 29798270
```

pxyscand est utilisé par le serveur IRC **undernet.org** pour détecter si les utilisateurs passent par un proxy.

```
92.62.43.77 - - [24/Apr/2010:16:34:42 -0700] "CONNECT 92.62.43.77:6667 HTTP/1.0" 301 - "-" "-"
-Pu0pQoAAQ4AAEHDCQ8AAAAA 240388
```

Le même type de test venant cette fois-ci de **proxyscanner.quakenet.org**.

Le fichier **apache2/www-access.log** ne contient que des références à WordPress. Mais nous retrouvons certaines adresses IP vues précédemment :

- 201.229.176.217 - Apr 19 11:03:44

```
201.229.176.217 - - [19/Apr/2010:11:19:00 -0700] "GET /wp-
content/themes/optimize/css/style.css HTTP/1.1" 200 20673 "http://24.4.108.196/" "Mozilla/5.0
(Windows; U; Windows NT 5.1; es-ES; rv:1.9.0.19) Gecko/2010031422 Firefox/3.0.19" -
r@uVgoAAQ4AAEP3EbYAAAAD 851
```

- 190.167.70.87 - Apr 19 11:15:26

```
190.167.70.87 - - [19/Apr/2010:11:17:04 -0700] "GET /wp-content/themes/optimize/css/style.css
HTTP/1.1" 200 20673 "http://24.4.108.196/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US)
AppleWebKit/532.5 (KHTML, like Gecko) Chrome/4.1.249.1045 Safari/532.5"
99b2PQoAAQ4AAEP4ElYAAAAE 395
```

- 190.166.87.164 - Apr 19 22:37:24, le pirate ayant créé le compte **dhg** est venu consulter le site WordPress :

```
190.166.87.164 - - [21/Apr/2010:18:21:36 -0700] "GET /wp-content/themes/optimize/css/style.css
HTTP/1.1" 200 20673 "http://24.4.108.196/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US)
AppleWebKit/532.5 (KHTML, like Gecko) Chrome/4.1.249.1059 Safari/532.5"
IbqzmQoAAQ4AAARf7@wAAAAJ 1230
```

7. ÉCHEC DE L'ANONYMISATION

7.1 UTILISATEUR USER2

Le nom des comptes a été clairement modifié, mais ce nom a aussi été changé dans l'adresse **jp.user2pastoreinc.com**.

```
Apr 23 20:10:01 app-1 sshd[19886]: Invalid user escape from 173.9.147.165
Apr 23 20:10:01 app-1 sshd[19886]: pam_unix(sshd:auth): check pass; user unknown
Apr 23 20:10:01 app-1 sshd[19886]: pam_unix(sshd:auth): authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=jp.user2pastoreinc.com
```

```
Apr 23 20:10:03 app-1 sshd[19886]: Failed password for invalid user escape from 173.9.147.165
port 59177 ssh2
```

L'adresse IP inverse de 173.9.147.165 est **jp.jonpastoreinc.com**. Le vrai nom du compte **user2** est jon. De même, il y a eu une tentative d'intrusion sur le compte **jonathan** et non **user2athan**:

```
Apr 24 12:59:17 app-1 sshd[25041]: Invalid user user2athan from 8.12.45.242
```

D'après les logs, seule l'adresse 71.132.129.212 a essayé de se connecter sur ce compte :

```
[kmaster@adsl sanitized_log]$ grep "password for user2" auth.log
Mar 18 11:38:05 app-1 sshd[10156]: Failed password for user2 from 71.132.129.212 port 34624
ssh2
Mar 18 11:38:10 app-1 sshd[10156]: Failed password for user2 from 71.132.129.212 port 34624
ssh2
Mar 18 11:38:43 app-1 sshd[10156]: Failed password for user2 from 71.132.129.212 port 34624
ssh2
Mar 18 11:38:59 app-1 sshd[10158]: Failed password for user2 from 71.132.129.212 port 34333
ssh2
Mar 18 11:39:50 app-1 sshd[10158]: Accepted password for user2 from 71.132.129.212 port 34333
ssh2
Mar 18 11:40:56 app-1 sshd[10200]: Accepted password for user2 from 71.132.129.212 port 40961
ssh2
Mar 18 11:41:43 app-1 sshd[10224]: Accepted password for user2 from 71.132.129.212 port 41661
ssh2
Mar 18 11:51:31 app-1 sshd[10294]: Accepted password for user2 from 71.132.129.212 port 41296
ssh2
Mar 23 18:02:29 app-1 sshd[7224]: Accepted password for user2 from 71.132.129.212 port 46820
ssh2
```

Nous avons considéré que cette adresse était celle d'un utilisateur légitime, car le mot de passe du compte a été modifié juste après les premiers essais infructueux.

7.2 PACKET & DHG

L'adresse 190.166.87.164 est connue pour avoir créé le compte **packet** ayant les droits **root** et le compte **dhg**, et s'être connecté sur ce dernier, mais ce pirate a laissé d'autres traces :

```
Apr 20 19:18:45 app-1 sshd[32210]: Accepted password for dhg from 190.166.87.164 port 58276
ssh2
Apr 20 20:00:26 app-1 sshd[32333]: reverse mapping checking getaddrinfo for
164.87.166.190.f.sta.codetel.net.do [190.166.87.164] failed - POSSIBLE BREAK-IN ATTEMPT!
Apr 20 20:00:26 app-1 sshd[32333]: Invalid user daniel from 190.166.87.164
```

Il utilise sans doute le compte **daniel** sur son ordinateur, il doit s'agir de son prénom. Google permet d'identifier plusieurs personnes ayant ce prénom et utilisant ce pseudo.

L'utilisateur **dhg** a installé les programmes :

- **psybnc** ;

- **eggdrop** permettant de rester connecter sur IRC.

```
Apr 19 23:21:08 app-1 sudo:      dhg : user NOT in sudoers ; TTY=pts/1 ; PWD=/home/dhg/psybnc-
linux/psybnc ; USER=root ; COMMAND=alien lsb-build-4.0.9-2.src.rpm
Apr 19 23:24:30 app-1 sudo:      dhg : user NOT in sudoers ; TTY=pts/1 ; PWD=/home/dhg/psybnc-
linux/psybnc ; USER=root ; COMMAND=root
Apr 19 23:25:00 app-1 sudo:     root : TTY=pts/1 ; PWD=/home/dhg/psybnc-linux/psybnc ;
USER=root ; COMMAND=/usr/bin/apt-get update
Apr 19 23:34:35 app-1 sudo:     root : TTY=pts/1 ; PWD=/home/dhg/eggdrop1.6.19 ; USER=root ;
COMMAND=/usr/bin/apt-get install tcl8.4 tk8.4
Apr 19 23:37:57 app-1 sudo:     root : TTY=pts/1 ; PWD=/home/dhg/eggdrop1.6.19 ; USER=root ;
COMMAND=/usr/bin/apt-get install tcl8.5-dev
```

```
Apr 19 23:38:18 app-1 sudo:     root : TTY=pts/1 ; PWD=/home/dhg/eggdrop1.6.19 ; USER=root ;
COMMAND=/usr/bin/apt-get install eggdrop
```

L'examen de **apt/term.log** indique que **eggdrop** a bien été installé sous forme de package :

```
Log started: 2010-04-19 23:38:27
Selecting previously deselected package eggdrop-data.
(Reading database ... 25259 files and directories currently installed.)
Unpacking eggdrop-data (from .../eggdrop-data_1.6.18-1.1ubuntu1_all.deb) ...
Selecting previously deselected package eggdrop.
Unpacking eggdrop (from .../eggdrop_1.6.18-1.1ubuntu1_amd64.deb) ...
Setting up eggdrop-data (1.6.18-1.1ubuntu1) ...
Setting up eggdrop (1.6.18-1.1ubuntu1) ...
Log ended: 2010-04-19 23:38:29
```

7.3 NOM DE DOMAINE

Le fichier **auth.log** contient une trace des commandes lancées via **sudo**. Une clé pour un certificat SSL a été créée pour le domaine **logggly.org** puis **domain.org**.

```
Mar 18 11:27:11 app-1 sudo:   user1 : TTY=pts/0 ; PWD=/etc/apache2 ; USER=root ;
COMMAND=/usr/bin/openssl req -new -key logggly.org.key -out logggly.org.csr
Mar 18 11:27:19 app-1 sudo:   user1 : TTY=pts/0 ; PWD=/etc/apache2 ; USER=root ;
COMMAND=/usr/bin/openssl req -new -key domain.org.key -out logggly.org.csr
```

Le domaine considéré était **loggly.org**, mais il a été mal orthographié en **logggly.org** (trois g). De ce fait, lorsque les logs ont été anonymisés, le nom de domaine erroné est resté dans le fichier.

Ce honeypot n'est cependant plus en fonction.

```
loggly.org has address 10.0.20.11
```

7.4 ADRESSE IP

Lorsqu'un navigateur suit un lien, il indique dans le champ **HTTP Referrer** l'URL de la page précédente ou – lorsque l'adresse est accédée directement. Les logs Apache enregistrent cette information :

Le fichier **apache2/www-media.log** :

```
201.229.176.217 - - [19/Apr/2010:11:19:00 -0700] "GET /wp-
content/themes/optimize/css/style.css HTTP/1.1" 200 20673 "http://24.4.108.196/" "Mozilla/5.0
(Windows; U; Windows NT 5.1; es-ES; rv:1.9.0.19) Gecko/2010031422 Firefox/3.0.19" -
r@uVgoAAQ4AAEP3EbYAAAAD 851
```

Une feuille de style pour WordPress est récupérée, elle a été listée par **http://24.4.108.196**. Nous venons probablement de récupérer l'adresse IP du honeypot. Cette adresse IP est listée dans un des exemples se trouvant sur **http://wiki.loggly.com/devic_apis**. Nous retombons sur une variante (autre extension) du nom de domaine trouvé précédemment.

8. Q & A

Reprenons les questions du challenge.

1. Le système a-t-il été compromis et quand ? En êtes-vous sûr ?

Le serveur a été compromis à plusieurs reprises via des accès SSH sur le compte **root**. Aucun doute n'est possible.

2. Si le serveur a été compris, quelle méthode a été utilisée ?

On observe de nombreuses tentatives rapprochées de connexions via SSH : attaques par force brute sur le compte **root**. Certains attaquants ont testé d'autres comptes, mais sans succès. On peut cependant penser que certains attaquants ont réussi lors de la première tentative!

3. Combien d'attaquants ont échoué ? Si certains ont réussi, combien sont-ils ? Combien ont arrêté après le premier succès ?

Sur les 41 adresses IP d'attaquants, nous avons 31 attaquants n'ayant jamais réussi à rentrer, et donc, 10 pirates ayant réussi.

Avec les critères énoncés, il y aurait 12 adresses IP d'utilisateurs légitimes, mais il est difficile de se prononcer sur 7 adresses IP.

Certains attaquants ont continué leur attaque par force brute alors qu'ils avaient réussi à trouver le mot de passe, c'est le cas des 6 attaquants suivants :

- 219.150.161.20 ;

- 222.66.204.246 ;

- 121.11.66.70 ;

- 222.169.224.197 ;

- 122.226.202.12 ;

- 61.168.227.12.

Petit bug ou problème de paramétrage dans l'outil d'attaque ? Difficile à dire.

4. Que s'est-il passé lors de l'attaque par brute force ?

- backdoor sur SSH ;

- installation de eggdrop ;

- installation de psybnc ;

- fichier secure vidé, un pirate a voulu effacer ces traces ?

5. D'après les logs d'authentification, combien d'attaques par force brute ont eu lieu ?

En comptant les 31 attaquants n'ayant pas réussi à se connecter et les 6 IP ayant réussi à se connecter qui ont au moins 5 échecs de connexion et plus d'échecs de connexion que de réussites, nous obtenons 37 adresses IP.

6. Quelle est la chronologie des événements ? L'horodatage est-il fiable ?

NTP est utilisé sur le serveur, il montre à un moment une erreur de plusieurs heures. Les logs étaient modifiables par les pirates. Cependant, on peut estimer l'horodatage comme assez fiable après la rectification initiale de l'heure.

7. Y a-t-il d'autres choses suspectes dans les logs ? D'autres problèmes de configuration ?

Les logs du serveur web Apache montrent des tentatives d'utilisation du serveur comme proxy, mais elles ont échoué.

8. Est-ce que les attaques ont utilisé des outils les automatisant ? Lesquels ?

Il est certain que les attaques par brute force automatisent les tentatives de connexion, et que sans outil, il n'est pas concevable de réaliser autant de tentatives de connexion en aussi peu de temps. Cependant, je n'ai pas trouvé d'éléments pour identifier les outils utilisés.

9. Quel était le but des attaquants ?

En dehors de l'utilisation du serveur pour accéder à des channels IRC, il est envisageable que le but soit de lancer des campagnes de spams ou autres activités associées à des botnets.

Question bonus. Qu'auriez-vous fait pour éviter l'attaque ?

Pour éviter l'attaque, il aurait suffi d'utiliser des mots de passe non triviaux.

En complément des mots de passe non triviaux, on peut utiliser le module **`recent`** de iptables pour limiter le nombre de connexions venant d'une même IP dans le temps. Comme des attaquants semblent avoir trouvé le mot de passe lors des

premiers essais, cela n'aurait pas été très efficace cependant. La vraie solution reste donc ici l'utilisation de mots de passe non triviaux.

Pour plus de sécurité, il est possible de forcer l'authentification par clé au niveau de la configuration de SSH et de filtrer les adresses IP autorisées à se connecter.

CONCLUSION

Si ce travail d'analyse de logs a permis d'identifier la majorité des pirates, c'est surtout parce que la majorité des fichiers de logs n'ont pas été effacés par ces pirates. Il conviendrait de centraliser les logs pour plus de sécurité. Malgré cette analyse, nous n'avons pas pu découvrir la majorité des actions effectuées par les utilisateurs, légitimes ou non. Il aurait été intéressant de connaître toutes les commandes lancées par `root`.