How to Install and Run AIDE (Advanced Intrusion Detection Engine)

Motivation

Discovering that a system has been compromised is crucial in uncovering Trojans and Advanced Persistent Threats (APT). There is no way to know that this has occurred unless you are running some type of tool that captures a hash of the file and rechecks those hashes periodically and automatically alerts the administrators of the system(s) of the potential attack [I have to say potential because the reality is that Big Bubba, the junior admin could have made a change without alerting the group].

Test environment layout

I used CentOS 6.5 setup with Vagrant.

```
Hard Drive: 40GB
Memory:     4GB
CPUs:       2
```

Setup and deployment

Elevate user privileges to root and execute the following command:

```
yum install aide --assumeyes
```

In the appendix is the modified /etc/aide.conf file which controls how aide runs. Adjust this to your environment's needs. I would also set the immutable bit on the file /etc/aide.conf to prevent other admins from messing with the file:

```
chattr +i /etc/aide.conf
```

Once that is done, run (to initialiaze the database):

```
aide --init
```

Move the database file to permanent position:

```
mv /var/lib/aide/aide.db.new.gz  /var/lib/aide/aide.db.gz
```

Then, to manually check the filesystem, run:

```
aide --check
```

To setup automated emails, run:

```
yum install mailx.x86_64 -y
export EDITOR=vi
crontab -e
# Daily AIDE integrity check run
0 0 * * * /usr/sbin/aide --check | /bin/mailx -s "$HOSTNAME - \
Daily AIDE integrity check run" your.admin@company.com
```

To have the system update the database:

```
aide -update
```

To burn the database to a CDROM, run:

```
yum install cdw.x86_64 cdrskin.x86_64 --assumeyes
mkisofs -V Aide_DB$(date +%F) -J -R -o /root/aide.iso /var/lib/aide/
cdrecord -v -eject /root/aide.iso
```

Conclusion

AIDE is an amazing open-source version to its paid cousin “Tripwire”. What it lacks is Enterprise capabilities in the sense of having a central management console that monitors all of the servers with monitoring and alerting capabilities. This could be achieved by setting up one server that has the root SSH key and uses SCP to move the database to the server, run the job, send an alert, remove the database and close the connection. You could then take this server offline when the jobs are not running. The trick with this is that you have to trust implicitly the admin of this server and his use of the root SSH key to access all of your servers internally. The other component that you have to balance is the time it takes to modify and maintain all of the config files (/etc/aide.conf) for every server. I believe these could be easily grouped, e.g. database servers (postgresql, mysql, oracle), web servers (Apache, Tomcat, JBoss), DNS servers, and other miscellaneous servers. This would cut down the number of configurations that an enterprise has to maintain.

Appendix

References:

http://www.techrepublic.com/article/use-aide-to-help-detect-a-compromised-system/

http://www.linux.com/learn/tutorials/386908-weekend-project-intrusion-detection-on-linux-with-aide

http://en.wikipedia.org/wiki/Advanced_persistent_threat

http://aide.sourceforge.net/

http://www.tripwire.com/

/etc/aide.conf:

```
# Example configuration file for AIDE.

@@define DBDIR /var/lib/aide
@@define LOGDIR /var/log/aide

# The location of the database to be read.
database=file:@@{DBDIR}/aide.db.gz

# The location of the database to be written.
#database_out=sql:host:port:database:login_name:passwd:table
#database_out=file:aide.db.new
database_out=file:@@{DBDIR}/aide.db.new.gz

# Whether to gzip the output to database
gzip_dbout=yes

# Default.
verbose=20

report_url=file:@@{LOGDIR}/aide.log
report_url=stdout
#report_url=stderr
#NOT IMPLEMENTED report_url=mailto:root@foo.com
#NOT IMPLEMENTED report_url=syslog:LOG_AUTH

# These are the default rules.
#
#p:      permissions
#i:      inode:
#n:      number of links
#u:      user
#g:      group
#s:      size
#b:      block count
#m:      mtime
#a:      atime
#c:      ctime
```

```
#S:      check for growing size
#acl:           Access Control Lists
#selinux        SELinux security context
#xattrs:        Extended file attributes
#md5:    md5 checksum
#sha1:   sha1 checksum
#sha256:        sha256 checksum
#sha512:        sha512 checksum
#rmd160: rmd160 checksum
#tiger:  tiger checksum

#haval:  haval checksum (MHASH only)
#gost:   gost checksum (MHASH only)
#crc32:  crc32 checksum (MHASH only)
#whirlpool:     whirlpool checksum (MHASH only)

#R:             p+i+n+u+g+s+m+c+acl+selinux+xattrs+md5
#L:             p+i+n+u+g+acl+selinux+xattrs
#E:             Empty group
#>:             Growing logfile p+u+g+i+n+S+acl+selinux+xattrs

# You can create custom rules like this.
# With MHASH...
# ALLXTRAHASHES = sha1+rmd160+sha256+sha512+whirlpool+tiger+haval+gost+crc32
ALLXTRAHASHES = sha1+rmd160+sha256+sha512+tiger
# Everything but access time (Ie. all changes)
EVERYTHING = R+ALLXTRAHASHES

# Sane, with multiple hashes
# NORMAL = R+rmd160+sha256+whirlpool
NORMAL = R+rmd160+sha256

## Custom names:
SizeOnly = s+b
SizeAndChecksum = s+b+md5+sha1
ReallyParanoid = p+i+n+u+g+s+b+m+a+c+md5+sha1+rmd160+tiger

# For directories, don't bother doing hashes
DIR = p+i+n+u+g+acl+selinux+xattrs

# Access control only
PERMS = p+i+u+g+acl+selinux

# Logfile are special, in that they often change
LOG = >

# Just do md5 and sha256 hashes
LSPP = R+sha256

# Some files get updated automatically, so the inode/ctime/mtime change
# but we want to know when the data inside them changes
DATAONLY =  p+n+u+g+s+acl+selinux+xattrs+md5+sha256+rmd160+tiger

# Next decide what directories/files you want in the database.

/boot   ReallyParanoid
/bin    ReallyParanoid
```

```
/sbin   ReallyParanoid
/lib    ReallyParanoid
/lib64  ReallyParanoid
/opt    ReallyParanoid
/usr    ReallyParanoid
/root   ReallyParanoid
# These are too volatile
!/usr/src
!/usr/tmp

# Check only permissions, inode, user and group for /etc, but
# cover some important files closely.
/etc    SizeAndChecksum
!/etc/mtab
# Ignore backup files
!/etc/.*~
/etc/exports  NORMAL
/etc/fstab    NORMAL
/etc/passwd   NORMAL
/etc/group    NORMAL
/etc/gshadow  NORMAL
/etc/shadow   NORMAL
/etc/security/opasswd   NORMAL

/etc/hosts.allow   NORMAL
/etc/hosts.deny    NORMAL

/etc/sudoers NORMAL
/etc/skel NORMAL

/etc/logrotate.d NORMAL

/etc/resolv.conf DATAONLY

/etc/nscd.conf NORMAL
/etc/securetty NORMAL

# Shell/X starting files
/etc/profile NORMAL
/etc/bashrc NORMAL
/etc/bash_completion.d/ NORMAL
/etc/login.defs NORMAL
/etc/zprofile NORMAL
/etc/zshrc NORMAL
/etc/zlogin NORMAL
/etc/zlogout NORMAL
/etc/profile.d/ NORMAL
/etc/X11/ NORMAL

# Pkg manager
/etc/yum.conf NORMAL
/etc/yumex.conf NORMAL
/etc/yumex.profiles.conf NORMAL
/etc/yum/ NORMAL
/etc/yum.repos.d/ NORMAL

/var/run/utmp LOG
```

```
!/var/log/.*
!/var/spool/.*

# This gets new/removes-old filenames daily
!/var/log/sa
# As we are checking it, we've truncated yesterdays size to zero.
!/var/log/aide.log

# LSPP rules...
# AIDE produces an audit record, so this becomes perpetual motion.
# /var/log/audit/ LSPP
/etc/audit/ LSPP
/etc/libaudit.conf LSPP
/usr/sbin/stunnel LSPP
/var/spool/at LSPP
/etc/at.allow LSPP
/etc/at.deny LSPP
/etc/cron.allow LSPP
/etc/cron.deny LSPP
/etc/cron.d/ LSPP
/etc/cron.daily/ LSPP
/etc/cron.hourly/ LSPP
/etc/cron.monthly/ LSPP
/etc/cron.weekly/ LSPP
/etc/crontab LSPP
/var/spool/cron/root LSPP

/etc/login.defs LSPP
/etc/securetty LSPP
/var/log/faillog LSPP
/var/log/lastlog LSPP

/etc/hosts LSPP
/etc/sysconfig LSPP

/etc/inittab LSPP
/etc/grub/ LSPP
/etc/rc.d LSPP

/etc/ld.so.conf LSPP

/etc/localtime LSPP

/etc/sysctl.conf LSPP

/etc/modprobe.conf LSPP

/etc/pam.d LSPP
/etc/security LSPP
/etc/aliases LSPP
/etc/postfix LSPP

/etc/ssh/sshd_config LSPP
/etc/ssh/ssh_config LSPP

/etc/stunnel LSPP
```

```
/etc/vsftpd.ftpusers LSPP
/etc/vsftpd LSPP

/etc/issue LSPP
/etc/issue.net LSPP

/etc/cups LSPP

# With AIDE's default verbosity level of 5, these would give lots of
# warnings upon tree traversal. It might change with future version.
#
#=/lost\+found    DIR
#=/home           DIR

# Ditto /var/log/sa reason...
!/var/log/and-httpd

# Admins dot files constantly change, just check perms
/root/\..* PERMS
```