CERT/CC

Mongolian Cyber Emergency Response Team / Coordination Center

Practical Recording of All TTY Sessions

Ts. Ganbold
MNCERT/CC advisor
FreeBSD committer

MNSEC 2017

About me

- FreeBSD user/sysadmin for over 18 years
- FreeBSD committer, doc – 2008, src – 2013
- MNCERT/CC advisor
- Worked for Government organizations, international project, ISP, mobile operator
- Do remote works related to FreeBSD – sysadmin, software development, consulting
- Hack ARM boards
- Football

Content

- Introduction
- Why TTY session recording?
- Recording TTY sessions
- Various TTY session recording tools, methods, problems
- OpenSSH server related configuration and considerations
- ttyrec, modifications, PoC patches
- Further improvement ideas

Introduction

It is simply not good enough to know who logged into your server and when. There is a strong need for a complete record of the user's activity, either to ensure that no unauthorized changes were made, or to meet compliance requirements such as PCI DSS.

Being able to identify who made specific changes and quickly reviewing them enables the sysadmin to react appropriately to a wide range of situations where you'd be otherwise guessing at what specifically they did. These actions include someone adding a backdoor user account, making an unauthorized configuration change, or even making a mistake they don't catch right away.

Why TTY session recording?

- Sometimes it is hard to find who did what on the system
- For security purpose – forensics
- PCI DSS requirements (10), compliance
- Can be good training material
- Catch mistakes and act accordingly

Recording TTY sessions

- Record shell activity
- TTY recording
- Run at login
- More deeper -> Audit framework

Various TTY session recording tools, methods, problems

Tools and Methods

- script command
- asciinema
- Via shell history, etc
- sudosh or sudosh2 etc
- tmux or ssh logging but on client side
- ttyrec

and maybe many more, but …

Problems

- Terminal window related problems - Control characters, cursors, colors etc, vi/vim, emacs, mc, etc.
- cat/more long files makes a log bigger
- System admins don't like to logout, they want to stay forever
- How to run at login ? User can kill the process
- Do we need to make changes in shell rc or profile file?
- Anyone can see plaintext logs

Problems

- Commands like script, ttyrec don't update utx/utmp database

  w command shows only ttyrec/script for the user, not the latest command

```
# w
10:37AM  up 10 days, 21:36, 3 users, load averages: 0.58, 0.44, 0.37
USER       TTY      FROM                              LOGIN@  IDLE WHAT
root       v0       -                                 14Sep17 10days -csh (csh)
test       pts/0    192.168.0.100                     10:36AM     - ttyrec
test       pts/2    192.168.0.100                     10:37AM     - w
```

- How about ssh?

OpenSSH server

Configuration and Considerations

- ForceCommand option - Run command or script when user login via ssh
- Problems - below cases should work
  - sftp
  - scp
  - some other commands – ls, etc

ttyrec, modifications, patches

ttyrec

- ttyrec (headers)
  - sec – unix timestamp
  - usec -- 0..999999 microseconds
  - len -- length of the payload
  - Actual payload
- BSD license

Modifications

- Option -l, that helps to reduce log (Antoine Amarilli <a3nm@a3nm.net>)
- Option -f, that will run shell if any crash happens (from same author as above)
- ssh command handling - scp, sftp etc

  - option -s to set sftp program

  - log ssh commands
- Set SETUID/SETGID bits and drop privileges later before running shell

  - chmod 4555 ttyrec

Modifications

- Update utx/utmp database – w command will show correct outputs

  - ptsname() – gets the path of the slave pseudoterminal

- Use setproctitle() to set process title - linux needs libbsd-dev package and setproctitle_init() call

```
# w
 2:19PM  up 7 days, 18:58, 4 users, load averages: 0.54, 0.38, 0.29
USER       TTY      FROM              LOGIN@  IDLE WHAT
tsgan      pts/0    192.168.0.100     10:47AM     - _su (csh)
tsgan      pts/1    192.168.0.100     10:49AM     - w
test       pts/2    192.168.0.100      2:19PM     - ttyrec: ssh-rec (ttyrec)
test       pts/3    192.168.0.100      2:19PM     - top
```

Modifications

- Log, key file name format: host-YYYY-MM-DD_HHhMMmSSs-user.ttyrec[.key]
- Encrypt log using stream cipher - AES-CTR (OpenSSL AES_ctr128_encrypt())
  - Option -p to set RSA public key to use
  - Generate random key and encrypt it to file using RSA public key
  - Use this generated random key to encrypt the log
  - Don't encrypt timestamps, len (ttyrec headers)
- ttydecrypt
  - Decrypt the encrypted key using RSA private key and then decrypt the log
  - Capsicumized for excersize (https://wiki.freebsd.org/Capsicum)

DEMO

Patches

- The PoC patch for FreeBSD is at:

  https://github.com/tsgan/ttyrec-patch

- Tested on FreeBSD 10.X, 11.X, Ubuntu 14.04 (with small changes to the patch)

Further improvement ideas

- Maybe use OpenSSL EVP_* functions for encrypt/decrypt (AESNI hardware acceleration etc.)
- Log to centralized logging server

  - Use/integrate fluentd/graylog

  - Or write separate daemon

  - …

Useful links etc.

https://github.com/mjording/ttyrec
http://asciinema.org
https://github.com/ConradIrwin/showterm
https://github.com/theonewolf/TermRecord
http://tty-player.chrismorgan.info/
https://github.com/jory/ttyplayer
https://github.com/yudai/gotty
https://github.com/krishnasrinivas/wetty
https://github.com/tmux-plugins/tmux-logging
https://github.com/tsl0922/ttyd
https://www.cyberciti.biz/faq/linux-unix-login-bash-shell-force-time-outs/
https://github.com/cloudposse/sudosh
http://a3nm.net/blog/ttyrex.html
https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation
https://wiki.openssl.org/index.php/EVP

Thank you for your attention

CERT/CC

Mongolian Cyber Emergency Response Team / Coordination Center