Linux Firejail

A MUUG Presentation

About Trevor Cordes

- UNIX-head since 1992 (SunOS > AIX > RH > Fedora)
- Fedora, PHP & Perl fan (wanna fight?)
- MUUG Vice-President
- STUG Past-President (defunct Atari ST club)
- Owner, Tecnopolis Enterprises
  - Celebrating 25 years in business March 1

Deja Vu?

- MUUG meeting: November 2007
- Shawn Wallbridge
- FreeBSD Jails & Solaris Zones
- But not really the same thing...

WHY

- Very complex programs (e.g. browsers)
- Security holes found seemingly every fortnight
- Other software you don’t trust
- Not from repos
- Windows programs with WINE

WHY – Browsers

Google » Chrome: CVSS Scores Betwe

Period 2004 - 02 - 05 2024 - 02 - 05

CVSS Score Range	Vulnerabilities
0-1	3
1-2	0
2-3	5
3-4	4
4-5	410
5-6	254
6-7	757
7-8	531
8-9	1049
9+	308
Total	3321

Weighted Average CVSS Score: 7.7

Mozilla » Firefox: CVSS Scores Betwee

Period 2004 - 02 - 05 2024 - 02 - 05

CVSS Score Range	Vulnerabilities
0-1	0
1-2	2
2-3	49
3-4	12
4-5	337
5-6	325
6-7	423
7-8	334
8-9	316
9+	744
Total	2542

Weighted Average CVSS Score: 7.8

(source: cvedetails.com)

WHY

- Nice if we could hide all sensitive data
- Deny access to unnecessary programs
- Just for the desired application
- Easily
- Like one extra word of typing
- Or one-time setup program
- And run with no performance penalty at all

WHY

- Must provide easy access to specified files
- Even in otherwise blocked directories
- Hide all of HOME
- But allow access to ~/.mozilla and ~/.cache/mozilla
- And Downloads

WHY not...?

- SELinux?
- Full virtualization?
- Containers?
- more on this later...

WHAT

- Firejail
- Firejail is a SUID sandbox program that reduces the risk of security breaches by restricting the running environment of untrusted applications using Linux namespaces and seccomp-bpf.
- It allows a process and all its descendants to have their own private view of the globally shared kernel resources, such as the network stack, process table, mount table.
- (source: man firejail)

Demo 0

- firejail firefox

WHAT

- The sandbox is lightweight, the overhead is low. There are no complicated configuration files to edit, no socket connections open, no daemons running in the background. All security features are implemented directly in Linux kernel and available on any Linux computer.
- (source: https://sourceforge.net/projects/firejail/)

WHAT

- Instead of adding daemons and other applications, it works by creating a restricted environment with its own set of solutions, running within user space and using features that are already a part of the Linux kernel, such as seccomp-bpf.
- (source: Linux Pro Magazine #189 article by Bruce Byfield)

WHAT

- Best for interactive users
- Desktop or command line
- Not daemons (but you can…)
- The supplied profiles are all for desktop/user programs

Uses

- I’ve used it for:
- Firefox
- Steam
- Windows programs with WINE: Studiotax, Goldwave
- NodeJS NPM
- Random non-repo software: Linux Pro Mag all-issues DVD archive searcher in Java (AWT, eww!)

WHAT

- Virtually no overhead
- Uses existing kernel facilities for everything
- Sets it all up; slight launch delay (0.5s)
- Then gets out of the way
- 3D games run full speed
- Virtually no RAM overhead
- Shared libraries still shared by host OS regardless of jail

Alternatives: Virtualization

- Full virtualization
- Overkill
- Too much time to create & spin up
- Too many resources, often pre-allocated
- Image disk space
- Reserved RAM space
- Continuous CPU overhead (4%+)

Alternatives: Virtualization

- Shared library sharing not always possible
- ksmd helps alleviate this
- Requires after-the-fact scans
- Software must inform the OS with madvise()
- Some virtualization systems do this

Alternatives: Containers

- Docker, FreeBSD jails (sort of), etc.
- Similar to firejail in some ways (namespaces)
- Don’t pre-allocate resources
- But…
- Waste disk space with copies of all required libraries, files
- No shared library savings (dissimilar versions)

Alternatives: SELinux

- Whole-system
- All root-controlled
- Difficult setup & tweak
- No ease-into-it transition
- Difficult to grok

Permissive by Default

- firejail would default to blocking nothing if you don’t give options or profile
- Firewall terminology: “default allow”
- But it provides auto-applied default profiles that do block what you want, especially if your filesystem layout follows LSB
- Whitelisting options do exist
- Could try a “default deny”

HOW

- **firejail *program***
- **Symlinks:**
  - PATH must list /usr/local/bin first
  - env |grep PATH
  - ln -s /bin/firejail /usr/local/bin/*program*

HOW – I said E.A.S.Y!

- firecfg
- Easy setup for desktop and command line launchers
- Run as root with no options, or --guide
- --clean like it never existed
- --fix fixes up .desktop files that may use full paths
- --list see what it did
- No worries: only affects /usr/local/bin

Demo 1

- The juicy stuff

Customizations

- Put your own profiles in:
- ~/.config/firejail/foo.profile
- ~/.config/firejail/foo.local
- /etc/firejail/foo.local
- Not existing /etc/firejail/*.profile files
- Or any location, then specify in launchers with --profile=

RTFM

- man firejail-profile

Interesting Options

- noroot
- noblacklist must come before a conflicting blacklist
- --debug-blacklists --debug-whitelists

Networking

- Full separate network namespace supported
- Optional
- Bridges, tap
- Its own firewall

Private Mode

- Quick & dirty sandbox
- firejail --seccomp --private *program*
- HOME will be completely hidden
- Any writes will be thrown away when jail quits

Under The Hood

- Namespaces
- Isolation of processes and the abstraction of resources that these processes use. The mount namespace lets you select which mountpoints are to be visible in the process group. The PID namespace abstracts the process IDs, assigning an ID of 1 to the first process within the process group.
- Others: network, IPC, user, etc.
- (source: Admin Magazine #66 / Matthias Wubbeling, paraphrased)

Under The Hood

- seccomp (bpf)
- ...is a computer security facility in the Linux kernel. seccomp allows a process to make a one-way transition into a "secure" state [in terms of blacklisting selected system calls.] (source: Wikipedia)
- To help creating useful seccomp filters more easily, the following system call groups are defined: @aio, @basic-io, @chown, @clock, @cpu-emulation, @debug, @default, @default-nodebuggers, @default-keep, @file-system, @io-event, @ipc, @keyring, @memlock, @module, @mount, @network-io, @obsolete, @privileged, @process, @raw-io, @reboot, @resources, @setuid, @swap, @sync, @system-service and @timer (source: man firejail)

Under The Hood

- caps – Linux Capabilities
- Linux divides the privileges traditionally associated with superuser into distinct units, known as capabilities, which can be independently enabled and disabled.
- CAP_AUDIT_CONTROL,CAP_CHOWN,CAP_SYS_NICE, etc.

Under The Hood

- caps
- Drops the “usual suspects” ones
- caps.drop all
- Drops everything
- Recommended if it works
- Overlaps with seccomp some

Is It Safe?

- suid programs are difficult to secure
- Many CVEs in 2016-2017
- When firejail first became popular
- Almost all were suid-based local user attacks
- Only 1-2 were sandbox escapes
- Very few CVEs since

Is It Safe?

- Stop suid local user attacks:
- /etc/firejail/firejail.users
- Or make a group who can run firejail
- chown root:firejailgroup /bin/firejail
- chmod 550
- Will break on next package update

Cool Advanced Features

- Run another distro’s userspace programs
- mkdir /debian
- debootstrap --arch=amd64 stable /debian
- firejail --chroot=/debian firefox
- uids must match

Firejail a User

- adduser --shell /bin/firejail joeblow
- Like a better restricted shell

File Transfer In/Out Of Jail

- --cat=pid filename
- --get
- --put
- --ls

Traffic Shaping

- Uses Linux tc traffic shaping
- Which is normally very complex
- firejail --bandwidth=mybrowser set eth0 80 20
- Download/upload

Even Daemons

- Nginx, etc.
- Exercise for the reader...

Thank you

- Inspiration from Linux Pro Magazine
- Articles in issues #173 & #189
- linuxpromagazine.com

Lastly

- nonewprivs
- Ensures children cannot elevate privileges via a suid program
- Defaults on if seccomp on