19 November 2014
Debian führt Systemd ein [colourized]
Why I dislike systemd

(Published 2015-06-14)

As a Linux sysadmin in the 2010s, it’s hard not to have an opinion on systemd. But what I find baffling about it is how divisive it is; nearly everyone (or at least the most vocal crowd) seems to either love it or hate it. When I tell people that systemd was the catalyst for my defection to OpenBSD last year, their usual reaction is to assume that I am part of the "hate it" group. Nope.

In truth, systemd itself was a very small part of the reason I jumped ship. Its introduction made me realise two important things. First, the design problems with modern Linux run deeper than any one piece of software, I just hadn’t noticed until I had a fresh one to learn. Second, and this is specific to Debian, the "universal operating system" mantra is fundamentally flawed;
Systemd is bad parsing and should feel bad

Systemd has a remotely exploitable bug in its DHCPv6 client. That means anybody on the local network can send you a packet and take control of your computer. The flaw is a typical buffer-overflow. Several news stories have pointed out that this client was rewritten from scratch, as if that were the moral failing, instead of reusing existing code. That's not the problem.

The problem is that it was rewritten from scratch without taking advantage of the lessons of the past. It makes the same mistakes all over again.

In the late 1990s and early 2000s, we learned that parsing input is a problem. The traditional ad hoc approach you were taught in school is wrong. It's wrong from an abstract theoretical point of view. It's wrong from the practical point of view, error prone...

[20], [14],
Broken by design: systemd
09 Feb 2014 19:56:09 GMT

Recently the topic of systemd has come up quite a bit in various communities in which I'm involved, including the mustl IRC channel and on the Busybox mailing list.

While the attitude towards systemd in these communities is largely negative, much of what I've seen has been either dismissable by folks in different circles as mere conservatism, or tempered by an idea that despite its flaws, "the design is sound". This latter view comes with the notion that systemd's flaws are fixable without scrapping it or otherwise incurring major costs, and therefore not a major obstacle to adopting systemd.

My view is that this idea is wrong: **systemd is broken by design**, and despite offering highly enticing improvements over legacy init systems, it also brings **major regressions** in terms of many of the areas Linux is expected to excel: security, stability, and not having to reboot to upgrade your system.

**The first big problem: PID 1**

On unix systems, PID 1 is special. Orphaned processes (including a special case: daemons which orphan themselves) get reparented to PID 1. There are also some special signal semantics with

[20], [14], [3],
Structural and semantic deficiencies in the systemd architecture for real-world service management, a technical treatise

by V.R.

Preface and disclaimer (♥)

You’re probably wide-eyed and gnawing at your teeth already.

I was finally tempted into writing this from a Hacker News discussion on “Debian Dropping the Linux Standard Base,” where some interest was expressed in reading an architectural critique of systemd.

To the best of my knowledge, this article - though it ultimately ended up more of a paper in article format, is the first of its kind. This is startling. It’s been over 5 years of systemd, and countless instances of religious warfare have been perpetrated over it, but even as it has become the dominant system in its area, there really hasn’t been a solid technical critique of it which actually dissects its low-level architecture and draws remarks from it.

In fact, much more worthwhile has been written on the systemd debate than on systemd itself.

[20], [14], [3], [23],
systemd is the best example of Suck.

There is a menace which is spreading like a disease throughout the Linux world, it is called systemd.

Short Historical Summary

systemd is a replacement for the standard init command, which normally runs as process id 1 on initialisation of a UNIX bootup. There has been a movement, especially around the Red Hat-related developers to copy Microsoft Windows and all of its features. Now this interpretation of how a userspace should look like is implemented and was introduced with big criticism and change in the Open Source world into many distributions. The [debacle in Debian](https://www.suckless.org) is the best example in how to not introduce such a changing technology into a distribution.

[20], [14], [3], [23], [12],
Systemd-redux

I figured it was about time for a followup on my systemd post. I’ve been meaning to do it for a while but time hasn’t allowed.

The end of Linux

Some people wrongly characterized this as some sort of hyperbole. It was not. Systemd IS changing what we know as Linux today. It remains to be seen if this is a good or bad thing but Linux is becoming something different than it was.

Linux is in for a rough few years

I do honestly believe this will end up being the start of a rocky period for Linux.
How to Crash Systemd in One Tweet

The following command, when run as any user, will crash systemd:

```
NOTIFY_SOCKET=/run/systemd/notify systemctl-daemon-notify
```

After running this command, PID 1 is hung in the pause system call. You can no longer start and stop daemons. inetd-style services no longer accept connections. You cannot cleanly reboot the system. The system feels generally unstable (e.g. ssh and su hang for 30 seconds since systemd is now integrated with the login system). All of this can be caused by a command that’s short enough to fit in a Tweet.
Hi

This is a heads up for a trivial systemd local root exploit, that was silently fixed in the upstream git as:

commit 06eac6bfe029804f296b065b3ee91e796e1cd0e
Author: ....
Date:  Fri Jan 29 23:36:08 2016 +0200

basic: fix touch() creating files with 07777 mode

mode_t is unsigned, so MODE_INVALID < 0 can never be true.

This fixes a possible DoS where any user could fill /run by writing to a world-writable /run/systemd/show-status.

The analysis says that is a "possible DoS", but its a local root exploit indeed. Mode 07777 also contains the suid bit, so files created by touch() are world writable suids, root owned. Such as /var/lib/systemd/timers/stamp-tstrim.timer that's found on a non-nosuid mount.

fit in a Tweet.

[6], [19],
systemd v233 and earlier fails to safely parse usernames starting with a numeric digit (e.g. "0day"), running the service in question with root privileges rather than the user intended.

Publish Date : 2017-07-07 Last Update Date : 2017-07-22

- CVSS Scores & Vulnerability Types

  CVSS Score 10.0

Confidentiality Impact Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication Not required (Authentication is not required to exploit the vulnerability)

[6], [19], [9]
... und es wird doch genutzt.
Also was ist systemd?

Moritz Müller    Philip Kaludercic

2019-11-11 — PASST
Teil I

Problem
Was ist PID 1?

nichts
Was ist PID 1?
Was ist PID 1?
Was ist PID 1? [24][21]

- System Services
- PID 1
- Hardware Configuration
- Zombie/Orphans
- Network Configuration
- User Input Systems
- Hardware Configuration
Was ist PID 1?

Diagramm: PID 1 ist verbunden mit den folgenden Systemen:
- User Input Systems
- System Services
- Hardware Configuration

Zitaten: [24][21]
Was ist PID 1?

PID 1

User Input Systems

System Services

Hardware Configuration

Network Configuration
Was ist PID 1?

- User Input Systems
- System Services
- Hardware Configuration
- Network Configuration
- Zombie/Orphans
Was ist PID 1?

PID 1

User Input Systems

System Services

Hardware Configuration

Network Configuration

Zombie/Orphans
int main() {
    int status;
    const char *shell[] = { "/bin/sh", NULL };
    if (!fork())
        execve(shell[0], shell, NULL);
    while (1)
        wait(&status);
}
In “init(1)”,

```plaintext
sys intrp
jms init1 " Fork the first child
    " connected to ttyin/ttyout
jms init2 " Fork the second child
    " connected to keyboard/display

1:
sys rmes " Wait for a child to exit
sad pid1
   jmp 1f " It was child 1, so jump to
    " 1f and restart it
sad pid2
...

login:
-1
sys intrp
sys open; password; 0 " Open the passwd file
lac d1
sys write; m1; m1s " Write \nlogin:" on
    " the terminal
jms rline " and read the user's
    " username
...
```

- PDP-7 Unix hatte nur einen Assembler \[1\]
- Wird später auf ein Shellscript ausgelagert \[2\]
- /etc/rc wird mit der Zeit erweitert \[21\]
In “init(1)”

Aus init(8) in Unix-7:

“When init comes up multiuser, it invokes a shell, with input taken from the file /etc/rc. This command file performs housekeeping like removing temporary files, mounting file systems, and starting daemons.”

• PDP-7 Unix hatte nur einen Assembler[1]
• Wird später auf ein Shellscrip ausgelagert[2]
In “init(1)”

```bash
#!/bin/bash

start() { ... }
stop() { ... }

case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  restart|reload)
    stop
    start
    ;;
  *)
    exit 1
esac
```

- PDP-7 Unix hatte nur einen Assembler[^1]
- Wird später auf ein Shellscript ausgelagert[^2]
- `/etc/rc` wird mit der Zeit erweitert[^21]
“Shell scripts tend to be slow, needlessly hard to read, very verbose and fragile.”

— Lennart Poettering[22], 2010
2010 Lennart Pottering (Red Hat) will init(8) überdenken[21][16]
2010 Lennart Pottering (Red Hat) will init(8) überdenken[21][16]

2011 Obwohl zunächst zögernd[7], wird systemd in Fedora umgesetzt[4][13]
2010 Lennart Pottering (Red Hat) will init(8) überdenken [21][16]

2011 Obwohl zunächst zögernd [7], wird systemd in Fedora umgesetzt [4][13]

2010- Das Projekt wird organisiert mittels Mailingslisten[30], IRC[8], auf freedesktop.org[25]
Wer?

2010  Lennart Pottering (Red Hat) will init(8) überdenken.[21][16]

2011  Obwohl zunächst zögernd[7], wird systemd in Fedora umgesetzt[4][13]

2010- Das Projekt wird organisiert mittels Mailingslisten[30], IRC[8], auf freedesktop.org[25]

2015- Entwicklung wird auf GitHub verlagert[15] systemd/systemd
2010 Lennart Pottering (Red Hat) will init(8) überdenken[21][16]

2011 Obwohl zunächst zögernd[7], wird systemd in Fedora umgesetzt[4][13]

2010- Das Projekt wird organisiert mittels Mailingslisten[30], IRC[8], auf freedesktop.org[25]

2015- Entwicklung wird auf GitHub verlagert[15] ... alte Strukturen bleiben aber immernoch aktiv.
Teil II

Lösung?
Komponenten von systemd
Komponenten von systemd

PID 1

systemd
Komponenten von systemd

PID 1

systemd

networkd
Komponenten von systemd

PID 1

logind

networkd

systemd
Komponenten von systemd

PID 1

- logind
- systemd
- networkd
- udevd
Komponenten von systemd

logind

systemd

networkd

udevd

... 69 Binaries
Beispiel: journald

systemd
Beispiel: journald

systemd

journald
Beispiel: **journald**\textsuperscript{[26]}

- systemd
- journald
- Kernel

Logs, Logs, Logs

/var/log/journald

+ Metadaten!

... in einem Binärformat.

```
grep(1)
```

also was dann?
Beispiel: `journald` [26]

- `systemd`
- `journald`
- `Kernel`
- `Boot`

Dienste, Logs, Logs, Logs

`/var/log/journal`

+ Metadaten!

... in einem Binärformat

`grep(1)`

also was dann?
Beispiel: journald

- systemd
- journald
- logs, logs, logs
- Dienste
- Kernel
- Boot

+ Metadaten!
- grep(1)

also was dann?
Beispiel: `journald`[26]

- `systemd`
- `journald`
- `/var/log/journal`
- Dienste
- Kernel
- Boot

Logs, logs, logs

Grepp (1) nicht möglich

so also was dann?
Beispiel: `journald`

- `systemd`
- `journald`
- `Kernel`
- `Dienste`
- `Boot`

`/var/log/journal`

+ `Metadaten!`

Logs, logs, logs

grep(1) nicht möglich... also was dann?
Beispiel: journald

systemd

journald

logs, logs, logs

/Dienste

Kernel

Boot

/var/log/journal

+ Metadaten!

... in einem Binärformat
Beispiel: `journald`[26]

systemd → `journald` → logs, logs, logs → Dienste

`journald` → logs → Kernel

`journald` → logs → Boot

`journald` → `/var/log/journal`

+ Metadaten!

... in einem Binärformat

`grep(1)`
Beispiel: `journald`\[26\]

- systemd → journald
- journald → /var/log/journal
- journald → Dienste
- journald → Kernel
- journald → Boot
- /var/log/journal

+ Metadaten!

... in einem Binärformat

`grep(1)`

`nicht möglich`
Beispiel: `journald`

- `systemd` -> `journald` -> `/var/log/journal`
  - `+ Metadaten!`
  - `... in einem Binärformat`
- `Dienste` -> `journald` -> `/var/log/journal`
- `Kernel` -> `journald` -> `/var/log/journal`
- `Boot` -> `journald` -> `/var/log/journal`
  - `logs, logs, logs`
- `grep(1)`
  - `nicht möglich`
  - `also was dann?`
Schnittstelle: journalctl

$ journalctl
Nov 04 14:56:46 faui04p sshd[32571]: Received disconnect from 10.21.3.183 port 57852:11: disconnected by user
Nov 04 14:56:46 faui04p sshd[32571]: Disconnected from user ig56ewaw 10.21.3.183 port 57852
Nov 04 14:56:55 faui04p dbus-daemon[32698]: [session uid=68120 pid=32696] Activating service name='org.a11y.Bus' requested by ':1.0' (uid=68120 pid=32693 comm="gitg ")
Nov 04 14:56:55 faui04p dbus-daemon[32698]: [session uid=68120 pid=32696] Successfully activated service 'org.a11y.Bus'
Nov 04 14:56:55 faui04p org.a11y.Bus[32698]: dbus-daemon[32705]: Activating service name='org.a11y.atspi.Registry' requested by ':1.0' (uid=68120 pid=32693 comm="gitg ")
Nov 04 14:56:55 faui04p org.a11y.Bus[32698]: dbus-daemon[32705]: Successfully activated service 'org.a11y.atspi.Registry'
Nov 04 14:56:55 faui04p org.a11y.Bus[32698]: SpiRegistry daemon is running with well-known name - org.a11y.atspi.Registry
Nov 04 14:56:55 faui04p dbus-daemon[32698]: [session uid=68120 pid=32696] Activating service name='ca.desrt.dconf' requested by ':1.2' (uid=68120 pid=32693 comm="gitg ")
...
Schnittstelle: `journalctl` ... zu viel

$ journalctl
Nov 04 14:56:46 faui04p sshd[32571]: Received disconnect from 10.21.3.183 port 57852:11: disconnected by user
Nov 04 14:56:46 faui04p sshd[32571]: Disconnected from user ig56ewaw 10.21.3.183 port 57852
Nov 04 14:56:55 faui04p dbus-daemon[32698]: [session uid=68120 pid=32696] Activating service name=org.a11y.Bus requested by ':1.0' (uid=68120 pid=32693 comm="gitg")
Nov 04 14:56:55 faui04p dbus-daemon[32698]: [session uid=68120 pid=32696] Successfully activated service 'org.a11y.Bus'
Nov 04 14:56:55 faui04p org.a11y.Bus[32698]: dbus-daemon[32705]: Activating service name='org.a11y.atspi.Registry' requested by ':1.0' (uid=68120 pid=32693 comm="gitg")
Nov 04 14:56:55 faui04p org.a11y.Bus[32698]: dbus-daemon[32705]: Successfully activated service 'org.a11y.atspi.Registry'
Nov 04 14:56:55 faui04p org.a11y.Bus[32698]: SpiRegistry daemon is running with well-known name - org.a11y.atspi.Registry
Nov 04 14:56:55 faui04p dbus-daemon[32698]: [session uid=68120 pid=32696] Activating service name='ca.desrt.dconf' requested by ':1.2' (uid=68120 pid=32693 comm="gitg")

...
Auf den Boot beschränken?

$ journalctl -b
Auf den Boot beschränken?
$ journalctl -b
Auf den vorigen Boot beschränken?
$ journalctl -b -1
Auf den Boot beschränken?

$ journalctl -b

Auf den vorigen Boot beschränken?

$ journalctl -b -1

Nur Fehler?

$ journalctl -p err
Auf den Boot beschränken?

$ journalctl -b

Auf den vorigen Boot beschränken?

$ journalctl -b -1

Nur Fehler?

$ journalctl -p err

Zeitlich beschränkt?

$ journalctl --since=2019-11-10
             --until="2019-11-11 23:59"
Auf den Boot beschränken?

$ journalctl -b

Auf den vorigen Boot beschränken?

$ journalctl -b -1

Nur Fehler?

$ journalctl -p err

Zeitlich beschränkt?

$ journalctl --since=2019-11-10

--until="2019-11-11 23:59"

Live-Log vom Apache-Prozess?

$ journalctl -f -u apache
Auf den Boot beschränken?
    $ journalctl -b
Auf den vorigen Boot beschränken?
    $ journalctl -b -1
Nur Fehler?
    $ journalctl -p err
Zeitlich beschränkt?
    $ journalctl --since=2019-11-10
                  --until="2019-11-11 23:59"
Live-Log vom Apache-Prozess?
    $ journalctl -f -u apache
Nach beliebigen Metadaten gefiltert?
    $ journalctl _UID=70 _COMM=pulseaudio
Vielen Dank für die Aufmerksamkeit!

Anregungen zur Diskussion:

- Versucht systemd zu viel auf einmal?
- Hat systemd es geschafft, die Linux-Welt zu standardisieren?
- Persönliche Erfahrungen?
Referenzen


Referenzen V


Referenzen VII


Referenzen VIII


[29] `systemd(1): systemd, init - systemd system and service manager`.
