Andrew Ayer

Sections

← Previous

About this Post

Date 2016-09-28
Time 19:14:27 (UTC)

Recent Posts

Archives

Subscribe

Atom Feed

Login

You are here: Andrew's Site → Blog → How to Crash Systemd in One Tweet

How to Crash Systemd in One Tweet

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

```
NOTIFY_SOCKET=/run/systemd/notify systemd-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.

Edit (2016-09-28 21:34): Some people can only reproduce if they wrap the command in a `while true` loop. Yay non-determinism!

The bug is remarkably banal. The above systemd-notify command sends a zero-length message to the world-accessible UNIX domain socket located at `/run/systemd/notify`. PID 1 receives the message and fails an assertion that the message length is greater than zero. Despite the banality, the bug is serious, as it allows any local user to trivially perform a denial-of-service attack against a critical system component.

The immediate question raised by this bug is what kind of quality assurance process would allow such a simple bug to exist for over two years (it was introduced in systemd 209). Isn't the empty string an obvious test case? One would hope that PID 1, the most important userspace process, would have better quality assurance than this. Unfortunately, it seems that crashes of PID 1 are not unusual, as a quick glance through the systemd commit log reveals commit messages such as:

- coredump: turn off coredump collection only when PID 1 crashes, not when journald crashes
- coredump: make sure to handle crashes of PID 1 and journald special
- coredump: turn off coredump collection entirely after journald or PID 1 crashed

Systemd's problems run far deeper than this one bug. Systemd is defective by design. Writing bug-free software is extremely difficult. Even good programmers would inevitably introduce bugs into a project of the scale and complexity of systemd. However, good programmers recognize the difficulty of writing bug-free software and understand the importance of designing software in a way that minimizes the likelihood of bugs or at least reduces their impact. The systemd developers understand none of this, opting to cram an enormous amount of unnecessary complexity into PID 1, which runs as root and is written in a memory-unsafe language.

Some degree of complexity is to be expected, as systemd provides a number of useful and compelling features (although they did not invent them; they were just the first to aggressively market them). Whether or not systemd has made the right trade-off between features and complexity is a matter of debate. What is not debatable is that systemd's complexity does not belong in PID 1. As Rich Felker explained, the only job of PID 1 is to execute the real init system and reap zombies. Furthermore, the real init system, even when running as a non-PID 1 process, should be structured in a modular way such that a failure in one of the riskier components does not bring down the more critical components. For instance, a failure in the daemon management code should not prevent the system from being cleanly rebooted.

In particular, any code that accepts messages from untrustworthy sources like systemd-notify should run in a dedicated process as an unprivileged user. The unprivileged process parses and validates messages before passing them along to the privileged process. This is called privilege separation and has been a best practice in security-aware software for over a decade. Systemd, by contrast, does text parsing on messages from untrusted sources, in C, running as root in PID 1. If you think systemd doesn't need privilege separation because it only parses messages from local users, keep in mind that in the Internet era, local attacks tend to acquire remote vectors. Consider Shellshock, or the presentation at this year's systemd conference which is titled "Talking to systemd from a Web Browser."

Systemd's "we don't make mistakes" attitude towards security can be seen in other places, such as this code from the `main()` function of PID 1:

```
/* Disable the umask logic */
if (getpid() == 1)
        umask(0);
```

Setting a umask of 0 means that, by default, any file created by systemd will be world-readable and -writable. Systemd defines a macro called `RUN_WITH_UMASK` which is used to temporarily set a more restrictive umask when systemd needs to create a file with different permissions. This is backwards. The default umask should be restrictive, so forgetting to change the umask when creating a file would result in a file that obviously doesn't work. This is called fail-safe design. Instead systemd is fail-open, so forgetting to change the umask (which has already happened twice) creates a file that works but is a potential security vulnerability.

The Linux ecosystem has fallen behind other operating systems in writing secure and robust software. While Microsoft was hardening Windows and Apple was developing iOS, open source software became complacent. However, I see improvement on the horizon. Heartbleed and Shellshock were wake-up calls that have led to increased scrutiny of open source software. Go and Rust are compelling, safe languages for writing the type of systems software that has traditionally been written in C. Systemd is dangerous not only because it is introducing hundreds of thousands of lines of complex C code without any regard to longstanding security practices like privilege separation or fail-safe design, but because it is setting itself up to be irreplaceable. Systemd is far more than an init system: it is becoming a secondary operating system kernel, providing a log server, a device manager, a container manager, a login manager, a DHCP client, a DNS resolver, and an NTP client. These services are largely interdependent and provide non-standard interfaces for other applications to use. This makes any one component of systemd hard to replace, which will prevent more secure alternatives from gaining adoption in the future.

Consider systemd's DNS resolver. DNS is a complicated, security-sensitive protocol. In August 2014, Lennart Poettering declared that "systemd-resolved is now a pretty complete caching DNS and LLMNR stub resolver." In reality, systemd-resolved failed to implement *any* of the documented best practices to protect against DNS cache poisoning. It was vulnerable to Dan Kaminsky's cache poisoning attack which was fixed in every other DNS server during a massive coordinated response in 2008 (and which had been fixed in djbdns in 1999). Although systemd doesn't force you to use systemd-resolved, it exposes a non-standard interface over DBUS which they encourage applications to use instead of the standard DNS protocol over port 53. If applications follow this recommendation, it will become impossible to replace systemd-resolved with a more secure DNS resolver, unless that DNS resolver opts to emulate systemd's non-standard DBUS API.

It is not too late to stop this. Although almost every Linux distribution now uses systemd for their init system, init was a soft target for systemd because the systems they replaced were so bad. That's not true for the other services which systemd is trying to replace such as network management, DNS, and NTP. Systemd offers very few compelling features over existing implementations, but does carry a large amount of risk. If you're a system administrator, resist the replacement of existing services and hold out for replacements that are more secure. If you're an application developer, do not use systemd's non-standard interfaces. There will be better alternatives in the future that are more secure than what we have now. But adopting them will only be possible if systemd has not destroyed the modularity and standards-compliance that make innovation possible.

Hi, I'm Andrew. I'm the founder of SSLMate, a service which automates your SSL certificate deployment. I also develop open source projects like git-crypt and titus.

I blog here about a variety of technology topics, including security, devops, IPv6, and reliable programming. If you liked this post, check out my other posts or subscribe to my Atom feed.

My email address is andrew@agwa.name. I'm AGWA at GitHub and @__agwa on Twitter.

Comments

The comments below are owned by whoever posted them. I am not responsible for them in any way.

i just ran it. nothing happened. everything works exactly the same as before

Comment 31756 | Posted on 2016-09-28 at 20:33:07 UTC by Reader me | Reply to This

Some people are having to run the command in a loop for the bug to trigger: https://github.com/systemd/systemd/issues/4234#issuecomment-250289253

I'm not sure why; I was able to consistently reproduce with a single call under systemd v215, v229, and v230, on Debian and Ubuntu.

Comment 31759 | Posted on 2016-09-28 at 21:32:28 UTC by Andrew Ayer | Reply to This

Thank you for providing tangible arguments as to why Systemd is a dangerous option.

Comment 31757 | Posted on 2016-09-28 at 21:09:27 UTC by Anonymous | Reply to This

Please email them on secalert <@> redhat.com and they will take your issue seriously.

Comment 31762 | Posted on 2016-09-29 at 00:13:40 UTC by Reader Anonymous | Reply to This

I found I had to use a "while true" loop around that before it would trigger, as it seems it doesn't reliably trigger with every attempt (curious what combination of circumstances trigger it?)

Just out of curiosity, why didn't you contact the RedHat security team to responsibly disclose this vulnerability? (the majority of systemd development work is funded by, and done by, RedHat employees)

Comment 31758 | Posted on 2016-09-28 at 21:09:44 UTC by Reader Twirrim | Reply to This

This article starts with a bug. Note, it's a pretty irrelevant one that doesn't appear to be reproducible on recent builds.

Then it brings up a few other debatable issues, and then it incorrectly generalizes that systemd security is terrible. I say debatable because for example parsing command line parameters isn't something people usually delegate to a separate process.

Unfortunately, it also ignores all the great systemd security features it has added for the average Linux user (such as private tmp, and cgroups), and in fact all the other great features it has.

Comment 31760 | Posted on 2016-09-28 at 22:00:54 UTC by Reader Keith Curtis | Reply to This

> parsing command line parameters isn't something people usually delegate to a separate process.

It isn't about parsing command line. It's about parsing text coming from an untrusted socket where anyone can write to, at runtime, in something that runs as PID 1. Not exactly the same thing.

> Unfortunately, it also ignores all the great systemd security features it has added for the average Linux user (such as private tmp, and cgroups), and in fact all the other great features it has.

What it's debatable is that systemd add those things. I use private-tmp, namespaces, cgroups and more... and i don't have systemd installed in my system.

And anyway, cgroups are NOT a security feature; they can help to get a better security, but they aren't a security measure.

Comment 31765 | Posted on 2016-09-29 at 00:44:19 UTC by Reader nextime | Reply to This

Private tmp (being one application of namespaces) and cgroups are features of the Linux kernel, not systemd.

Also, the bug is very reproducible. I just tried it in Ubuntu 16.04 LTS after apt upgrade. Worked very, well^Wbadly.

Comment 31788 | Posted on 2016-09-29 at 07:22:22 UTC by Reader Antti Laine | Reply to This

I couldn't reproduce it on Ubuntu 16.04 LTS 32 bit.

Comment 31817 | Posted on 2016-09-30 at 08:33:17 UTC by Anonymous | Reply to This

Um... I can repeat it on 7 fully patched. I'm not sure what you mean by irrelevant. Now the patch has been submitted. When will a CVE be released. Sticking our heads in the sand is so ... Microsofty .... Let's not follow the win98 crew on that aspect please.

Comment 31808 | Posted on 2016-09-29 at 19:58:35 UTC by Reader James | Reply to This

Why not report this kind of denial-of-service attack in a responsible disclosure manner?

Comment 31761 | Posted on 2016-09-28 at 23:12:05 UTC by Reader Stefan | Reply to This

and you know that it was not responsibly reported how?

Comment 31801 | Posted on 2016-09-29 at 13:56:29 UTC by Reader tim | Reply to This

this is the most comprehensive writeup I've yet seen illustrating why systemd is not just a bad idea, but a worse implementation. The very concept is the complete opposite of the past 40 years of UNIX wisdom; that so many (well-respected) systems design folk seem to have completely forgotten their history (and really, just common sense) is depressing indeed.

Comment 31763 | Posted on 2016-09-29 at 00:29:25 UTC by Anonymous | Reply to This

(intended to sign that post, doing so here.)

Comment 31764 | Posted on 2016-09-29 at 00:30:01 UTC by Reader Scott Francis | Reply to This

http://static.usenix.org/event/lisa05/tech/full_papers/adams/adams.pdf This must be what systemd people saw but failed to implement in sane way.

Comment 31787 | Posted on 2016-09-29 at 07:02:00 UTC by Reader Jussi Sallinen | Reply to This

Fully agree. I'm just wondering what will this time be the excuse for closing the issue/bug report without even admitting the bug exists, or that it is a bug.

Comment 31793 | Posted on 2016-09-29 at 10:43:40 UTC by Reader Matthias Koch | Reply to This

They fixed the bug (although it took them two tries to get it right). But it's not about the bug. The bug is just one example of systemd's terrible programming practices.

Comment 31812 | Posted on 2016-09-29 at 23:18:07 UTC by Andrew Ayer | Reply to This

I agree with a large part of this. The only things I disagree about:

1. As a user and hobbyist system admin, systemd has numerous advantages over sysv init(which was the only previous init system to be installed by default on most distributions). So I would not say it has "very few" advantages over other init systems (I haven't used the other more modern init systems enough to have an informed opinion there). I also think having one standard init system between most distributions is very nice, so the fragmentation of the previous non-sysv solutions annoyed me.

2. Systemd, by definition, does not use a "proprietary" DBUS API nor "proprietary" interfaces. See http://www.merriam-webster.com/dictionary/proprietary if you don't believe me. Anyone can legally and practically implement a systemd DNS service clone that communicates via d-bus exactly like the systemd one, if they wish. That being said, I agree that it can still be a problem since it forces others to either imitate systemd's interface or get applications to use something standardized instead/as well.

Comment 31766 | Posted on 2016-09-29 at 01:03:36 UTC by Reader Christopher W. Carpenter | Reply to This

1. You misread me. I acknowledge that systemd's init system functionality has compelling features, even if terribly implemented. It's the other functionality -- DNS, DHCP, NTP -- where it doesn't offer anything new.

2. Fair enough. I replaced "proprietary" with "non-standard," which is what I meant.

Comment 31767 | Posted on 2016-09-29 at 01:56:44 UTC by Andrew Ayer | Reply to This

> As a user and hobbyist system admin

You seem to be a better adapted sysadmin than most of the non-hobbyist sysadmin's I know!

Comment 31768 | Posted on 2016-09-29 at 02:07:32 UTC by Reader jampola | Reply to This

I am a user and a professional Systems Admin. I might well note that this is the very kind of thing that does nothing for me. And a whole lot to me. I'd like to spend less time working around the things it breaks, please.

Comment 31809 | Posted on 2016-09-29 at 20:00:57 UTC by Reader James | Reply to This

This article is fantastic. I've been screaming about the rotten state of Linux due to Poettewrong's sysd for months. I've moved everything over to devuan, or left wheezy in place. Oh and I like sysvinit. So there. I REALLY like having log files I can tail too.

Comment 31769 | Posted on 2016-09-29 at 05:21:37 UTC by Reader Rhy | Reply to This

Ok Andrew, it works, i tested it in Debian Stable and Ubuntu LTS 16.04, every systemd distro that relies all their init scripts in systemd will crash with that command with an unprivileged user, so my Debian Stable with old legacy init scripts is not vulnerable, new Ubuntu systemdized distros (16.04) will crash systemd, no new processes can be launched, launched processes can't restart/reload, you can't even shutdown or reboot gracefully... It's amazing but what about Responsible Disclosure?

Comment 31790 | Posted on 2016-09-29 at 08:59:08 UTC by Reader ferchunix | Reply to This

Slackware is not using this bullshit of systemd and working fine for many years now !

Comment 31791 | Posted on 2016-09-29 at 09:52:56 UTC by Reader freesys59 | Reply to This

Devuan (https://devuan.org/) is Debian 8 systemd-free

Comment 31796 | Posted on 2016-09-29 at 12:17:46 UTC by Reader Andrea Mistrali | Reply to This

This does not appear to apply to OpenSUSE 13.2, as the path /run/systemd/notify does not exist. Running the command systemd-notify "" on its own does not appear to trigger the behaviour.

I'm not sure why that path doesn't exist.

Comment 31792 | Posted on 2016-09-29 at 10:42:03 UTC by Reader Wolf | Reply to This

Can you provide a link to the bug report you submitted to the systemd team before writing this blog post?

Comment 31794 | Posted on 2016-09-29 at 10:53:26 UTC by Reader James J. | Reply to This

People don't seem to understand how assert works. In release builds `assert` should be no-op. https://github.com/lattera/glibc/blob/master/assert/assert.h#L44 First of all assert is a debug tool, not error handling mechanism. Secondly, exiting with abort signal isn't very good error handling mechanism for `systemd`.

On the other hand, arguing that this makes `systemd` defective is hypocritical considering that Go FAQ says

> Go doesn't provide assertions. They are undeniably convenient, but our experience has been that programmers use them as a crutch to avoid thinking about proper error handling and reporting. (https://golang.org/doc/faq#assertions)

Comment 31795 | Posted on 2016-09-29 at 11:45:12 UTC by Reader Juha Autero | Reply to This

systemd's assertions are enabled in release builds. However, systemd doesn't exit when an assertion fails. Instead it hangs in the `pause` system call. This prevents the whole system from crashing, although the system is left in a degraded state.

I would have to say that this bug supports Golang's position on assertions.

Comment 31813 | Posted on 2016-09-29 at 23:21:15 UTC by Andrew Ayer | Reply to This

:(){ :|: & };:

Comment 31797 | Posted on 2016-09-29 at 12:23:17 UTC by Anonymous | Reply to This

Nobody I know likes systemd, it's too intrusive in user space and goes way against the core Unix philosophy of doing one thing and doing it well.

The sooner systemd disappears out of the Unix/Linux eco system the better.

Thank you for exposing this bug. We tried it on our up-to-date RedHat 7.2 systems and sure enough they crashed very hard.

Now to get rid of Not-workManager.

Comment 31798 | Posted on 2016-09-29 at 12:25:17 UTC by Reader orlwrite | Reply to This

If you haven't read it, check out this issue for another questionable security decision:

https://github.com/systemd/systemd/issues/959

Comment 31802 | Posted on 2016-09-29 at 14:29:22 UTC by Reader gdfuego | Reply to This

You could think systemd is as dangerous as ISIS

Comment 31803 | Posted on 2016-09-29 at 16:31:16 UTC by Anonymous | Reply to This

I feel sorry for people using distributions with systemd ''baked in''. During my day job, I am actually one of them, since my place of work is a big user of either Red Hat or Centos.

And then I remember that my personal systems are running either BSD or Slackware. And I feel better about my life.

Comment 31805 | Posted on 2016-09-29 at 16:55:45 UTC by Reader Noryungi | Reply to This

In some parts of SystemD they seem to use "bad coding practices from textbook":

```
    } else if (startswith(option, "keyfile-offset=")) {
            if (safe_atou(option+15, &arg_keyfile_offset) < 0) {
                    log_error("keyfile-offset= parse failure, ignoring.");
                    return 0;
            }
```

Do you see "option+15"? "15" here is a magic number, denoting the length of "keyfile-offset=":

https://cgit.freedesktop.org/systemd/systemd/tree/src/cryptsetup/cryptsetup.c#n106

Comment 31806 | Posted on 2016-09-29 at 18:38:50 UTC by Reader Andrey Bergman | Reply to This

Send patch?

Comment 31810 | Posted on 2016-09-29 at 21:37:24 UTC by Reader Damjan | Reply to This

I don't use systemD, so I even can't test the patch.

Comment 31814 | Posted on 2016-09-30 at 02:52:56 UTC by Reader Andrey Bergman | Reply to This

I continue not to understand why so *many* high-profile distributions' Release Configuration Managers drank the Flavor-Aid.

And the bad part is not that they did, but that *because they did*, packages are no longer required to have initscripts for *sane* inits, meaning that you have no choice: you either run systemd (read: shoot yourself in the head), or you *build initscripts by hand for every package you want to run*.

The only thing worse is reading the replies from systemd partisans when it's explained to them why the choices systemd's design makes are bad engineering practice -- am I the only one who thinks the structure of these replies is way too similar to the replies one hears from Trump supporters when it's explained why *he* is bad engineering practice? :-)

Added this article to my list of "Why systemd is bad" articles; added the supporting commenters to my list of "people never to hire or recommend for responsible software engineering jobs".

Comment 31807 | Posted on 2016-09-29 at 18:48:13 UTC by Reader Baylink | Reply to This

FUCK systemd!!!

Comment 31811 | Posted on 2016-09-29 at 22:43:18 UTC by Reader Linus | Reply to This

Does not work.

It say: "Failed to notify init system"...

Comment 31815 | Posted on 2016-09-30 at 07:54:21 UTC by Reader funt | Reply to This

I'm a systemd maintainer and have published some of my thoughts in a response: https://www.facebook.com/notes/david-timothy-strauss/how-to-throw-a-tantrum-in-one-blog-post/10108242959884190

This isn't an official response from the project, just one from someone who is very familiar with the issues in question.

Comment 31816 | Posted on 2016-09-30 at 08:02:32 UTC by Reader David Strauss | Reply to This

Post a Comment

Your comment will be public. If you would like to contact me privately, please email me. Please keep your comment on-topic, polite, and comprehensible. Use the "Preview" button to make sure your comment is properly formatted. Name and email address are optional. If you specify an email address it will be kept confidential.

Post Comment

Your Name: (Optional; will be published)

Your Email Address: (Optional; will not be published)

Your Website: (Optional; will be published)

Comment:

- Blank lines separate paragraphs.
- Lines starting with ">" are indented as block quotes.
- Lines starting with two spaces are reproduced verbatim.
- Text surrounded by *asterisks* is *italicized*.
- Text surrounded by `back ticks` is `monospaced`.
- URLs are turned into links.

Post Preview