*sshole

so openssh on musl linux (alpine, openwrt, buildroot, yocto, and a lot of the docker images you've pulled) has had a little problem since 2016.

accounts that should never take a password at all (root, your key-only deploy user, ldap users, the default user on alpine's aws and azure images) take any password longer than 256 characters.

`yes, any password. yes.`

```
$ ssh root@alpine-cloud
```

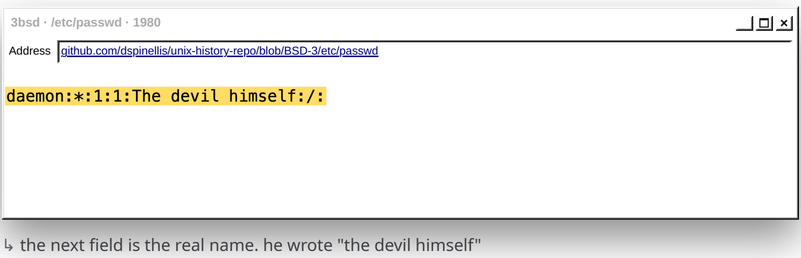
1980

the first star

back in old unix your password hash just sat in `/etc/passwd`, right next to your username. in a file everyone could read.

in 1980 bill joy put a `*` there for an account called `daemon`. nobody's supposed to ever log in as it.

a `*` isn't a hash, so no password can match it. unix still does this today


3bsd · /etc/passwd · 1980
Address
github.com/dspinellis/unix-history-repo/blob/BSD-3/etc/passwd

```
daemon:*:1:1:The devil himself:/:
```

↳ the next field is the real name. he wrote "the devil himself"

1990s

how sshd checks your password

ok so these days the hash lives in `/etc/shadow` and looks like `$6$abc$k2P9…`. the `$6$` means sha512, `abc` is the salt, the rest is the actual hash

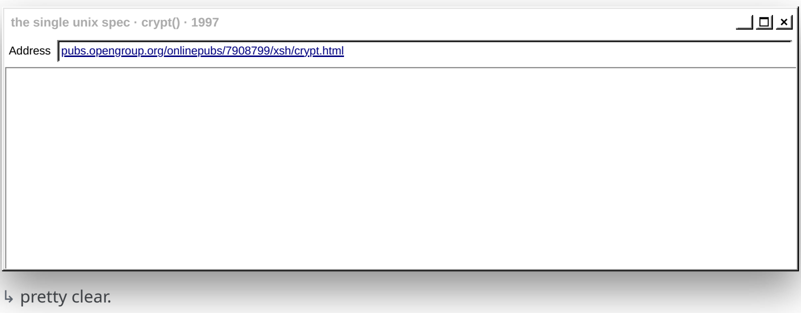
you type your password, sshd hashes it with that same salt and checks if it comes out identical. same = you're in. that's it

and if the field is just `*`? there's nothing to match. type whatever you want, it's never getting you in

1990s

the fine print

one more rule. if `crypt()` can't do its job, it returns NULL. like if the salt asks for some algorithm the system doesn't have, there's nothing to compute, so you get NULL instead of a hash


the single unix spec · crypt() · 1997
Address
pubs.opengroup.org/onlinepubs/7908799/xsh/crypt.html

↳ pretty clear.

except a lot of programs never check for NULL. they just compare whatever `crypt()` gives back with the stored field, hit the NULL, and crash

2012

crypt stops saying NULL

so musl (the libc under alpine) goes ok fine, `crypt()` just won't return NULL anymore. if it fails, it returns a single character instead. that character can never match a real password, so programs don't crash, they just see a wrong password

now guess which character they picked for that

`"?"`

musl · b3c4cc12 · aug 2012

Address git.musl-libc.org/cgit/musl/commit/?id=b3c4cc12

↳ "reportedly."

musl · b3c4cc12 · the diff

Address git.musl-libc.org/cgit/musl/commit/?id=b3c4cc12

↳ "should not happen" gone, NULL gone, `"*"` in

2012

three weeks later

then musl adds sha512 crypt. the thing is, sha512crypt gets slower the longer the password is, so someone could send one giant password and keep your server busy for a long time. so musl just caps it at 256 bytes and refuses anything longer

musl · 88bf5a8a · aug 29 2012

Address git.musl-libc.org/cgit/musl/commit/?id=88bf5a8a

↳ twenty days after the "*" commit btw

musl · crypt_sha512.c @ 88bf5a8a

Address git.musl-libc.org/cgit/musl/tree/src/misc/crypt_sha512.c?id=88bf5a8a

↳ so past 256 bytes crypt fails. and when crypt fails... it says `"*"`.

musl · crypt_sha512.c

Address git.musl-libc.org/cgit/musl/tree/src/crypt/crypt_sha512.c

↳ literally in the comment the whole time

2016

sshd borrows bob's salt

CVE-2016-6210. ask sshd about a user that doesn't exist and it says no instantly, cause there's nothing to hash. real users take a few ms longer. so just time it and you know who exists

the fix is kinda funny. no real hash to check? sshd grabs some other account's salt and hashes your password anyway, just to waste the same amount of time. still a no, just a slower no openssh-commits · jul 2016

Address lists.mindrot.org/pipermail/openssh-commits/2016-July/005428.html

↳ "for invalid users." and also, it turns out, real users with a `*`.

2019 → 2022

the fix for the fix

2019. turns out alpine's docker images had no root password. at all. CVE-2019-5021. alpine fixes it by locking root with `!`

but without pam, `!` also kills ssh keys. and cloud images log in with keys.

2022. alpine 3.16 switches root to `*`. still locked, keys work again

alpine aports · fbe926af · shipped in 3.16, 2022

Address gitlab.alpinelinux.org/alpine/aports/-/commit/fbe926af.patch

↳ narrator: it did not.

2016 → 2026

ten quiet years

why did nobody notice? well, maybe someone did. maybe someone pasted the wrong thing into a password prompt, half a slack message or a whole ssh key, and it just let them in. you've never pasted the wrong thing into a password prompt?? amazing. a fresh box with no passwords set isn't affected. a `*` in shadow needs a wrong try first. openssh tests on openbsd and glibc, musl doesn't test sshd, and most distros go through pam, which never even calls crypt for a `*`.

so as far as anyone knows, nobody noticed. as far as anyone knows.

some people got really close though

libc-alpha · debian #714219 · 2013

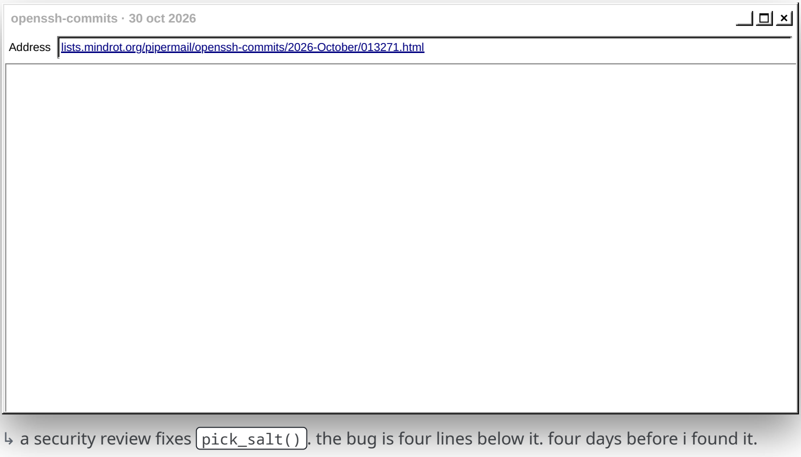
Address sourceware.org/legacy-ml/libc-alpha/2013-07/msg00046.html

↳ a debian dev asks glibc to copy musl's `"*"`. six years later debian moves to libxcrypt, which says `"*0"`, and closes the bug. one character away.

musl mailing list · dec 2024

Address openwall.com/lists/musl/2024/12/31/2

↳ this thread is about busybox `passwd` breaking on a 257 character password. same `"*"`, same limit. nobody says "sshd"


openssh-commits · 30 oct 2026

Address lists.mindrot.org/pipermail/openssh-commits/2026-October/013271.html

↳ a security review fixes `pick_salt()`. the bug is four lines below it. four days before i found it.

nov 2026

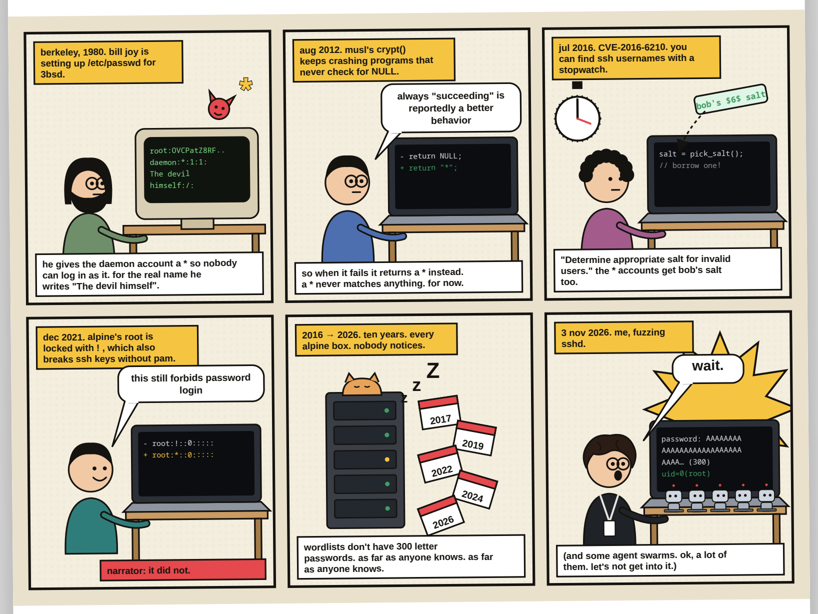
how i ran into it

i was messing around with openssh, mostly the boring parts where it talks to libc, like the password code. people don't really look at that stuff, they look at the crypto. so i built sshd every way i could think of (glibc, musl, pam, no pam, shadow, no shadow) and threw like 17,600 weird logins at it. weird hashes, broken accounts, passwords way too long

oh did i say "i"? sorry. i had agent swarms. a lot of them. let's not get into it

every build agreed on every login. except musl without pam. that one said yes to things it shouldn't

tl;dr


berkeley, 1980. bill joy is setting up /etc/passwd for 3bsd.
*

```
root:OVCPatZ8RF..
daemon:*:1:1:
The devil
himself:/:
```

he gives the daemon account a * so nobody can log in as it. for the real name he writes "The devil himself".
aug 2012. musl's crypt() keeps crashing programs that never check for NULL.
always "succeeding" is reportedly a better behavior

```
- return NULL;
+ return "*";
```

so when it fails it returns a * instead. a * never matches anything. for now.
jul 2016. CVE-2016-6210. you can find ssh usernames with a stopwatch.
bob's 6 salt

```
salt = pick_salt();
// borrow one!
```

"Determine appropriate salt for invalid users." the * accounts get bob's salt too.
dec 2021. alpine's root is locked with ! , which also breaks ssh keys without pam.
this still forbids password login

```
- root:!::0:::::
+ root:*::0:::::
```

narrator: it did not.
2016 → 2026. ten years. every alpine box. nobody notices.
Z
z
z
2017
2019
2022
2024
2026
wordlists don't have 300 letter passwords. as far as anyone knows. as far as anyone knows.
3 nov 2026. me, fuzzing sshd.
wait.

```
password: AAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAA… (300)
uid=0(root)
```

(and some agent swarms. ok, a lot of them. let's not get into it.)

try it

one debian box, one alpine box. type stuff. it all runs right here in your browser, nothing gets sent anywhere

password…

ozgun 256 × A 300 × A clear

256 1024

0 bytes

debian · glibc

crypt(pw, bob's salt)

—

alice's field

*

alpine · musl

crypt(pw, bob's salt)

—

alice's field

*

are you affected?

- ☐ is your sshd running on **musl**? (alpine, openwrt, buildroot, yocto)
- ☐ is it **not using pam**? alpine's `openssh-server` doesn't by default, or maybe you set `UsePAM no`
- ☐ are **password logins** still on? they are by default
- ☐ is there an account with just `*` as its password? key-only users, ldap users, root on alpine
- ☐ and some *other* account with a real password?

- ☐ **then that `*` account takes any password between 257 and 1024 characters**

on alpine's official aws and azure images that's the default `alpine` user, and it has passwordless `doas`. so it's root

on debian, ubuntu, fedora or anything else with glibc? going through pam? dropbear? openbsd? you're fine

what now

you

```
# update. openssh 10.6p2 fixes it.
$ apk upgrade openssh

# can't update yet? turn password logins off.
$ echo "PasswordAuthentication no" > /etc/ssh/sshd_config.d/00-no-passwords.conf
$ rc-service sshd restart
```

ONE MORE THING

did you get hacked? check.

root@your-box

see a line like that? an account with no password just took a password

found by **ozgun kultekin** at wakefield

we're hiring. wakefield.co/careers