


Rust, make me a sudo!

Ruben Nijveld, tweede golf

ruben@tweedegolf.com

 rnijveld

Ever used Linux/macOS/BSD?

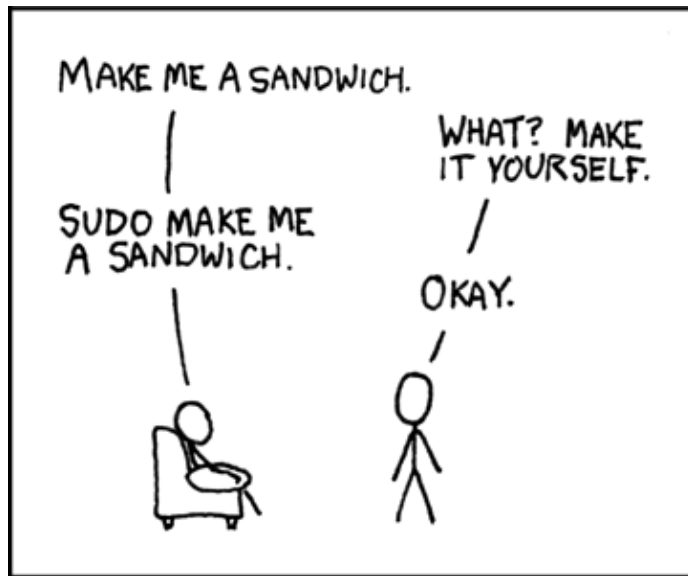
```
$ cat /root/hello.txt  
cat: /root/hello.txt: Permission denied
```

```
$ sudo cat /root/hello.txt  
[sudo] password for ruben:  
Hello world!
```

Ever used Linux/macOS/BSD? And for everyone else

```
$ cat /root/hello.txt  
cat: /root/hello.txt: Permission denied
```

```
$ sudo cat /root/hello.txt  
[sudo] password for ruben:  
Hello world!
```

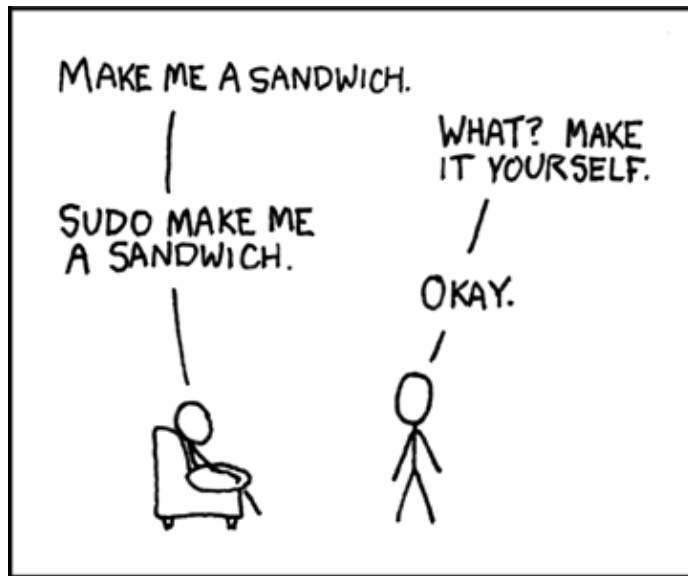
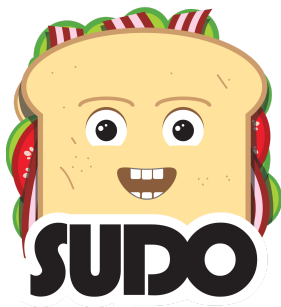


XKCD 149

Ever used Linux/macOS/BSD? And for everyone else

```
$ cat /root/hello.txt  
cat: /root/hello.txt: Permission denied
```

```
$ sudo cat /root/hello.txt  
[sudo] password for ruben:  
Hello world!
```



XKCD 149

We are rewriting sudo in Rust

Tweede golf and Ferrous Systems are building a drop in replacement for common use cases of sudo, commissioned by the ISRG, but why?

- Original sudo is over forty years old
- Was born before Linux was even a thought
- Legacy features, with legacy defaults
- Written in C, with lots of stringly typed interfaces
- A setuid program, executing as the owner (root) instead of the caller

We are rewriting sudo in Rust

Tweede golf and Ferrous Systems are building a drop in replacement for common use cases of sudo, commissioned by the ISRG, but why?

- Original sudo is over forty years old
- Was born before Linux was even a thought
- Legacy features, with legacy defaults
- Written in C, with lots of stringly typed interfaces
- A setuid program, executing as the owner (root) instead of the caller

Note: although some things may sound bad, sudo is a product of years of history and we think the original sudo maintainer actually does really great work. We should be happy to have him maintain it!

Memory safety

Of course you all know what I'm going to say

- Sudo has had its fair share of buffer overflows
- Modern tooling such as Rust can help us make a safer alternative
- Surely memory safety is the key feature that Rust offers

But wait, sudo actually does a lot more

Sudo allows a lot of customization, did you for example know that...

But wait, sudo actually does a lot more

Sudo allows a lot of customization, did you for example know that...

- User and group names are matched case insensitive by default?

But wait, sudo actually does a lot more

Sudo allows a lot of customization, did you for example know that...

- User and group names are matched case insensitive by default?
- You can use globs and regexes in specifying which commands you can run?

```
`ruben ALL = (ALL) cat /var/log/*`
```

But wait, sudo actually does a lot more

Sudo allows a lot of customization, did you for example know that...

- User and group names are matched case insensitive by default?
- You can use globs and regexes in specifying which commands you can run?

```
`ruben ALL = (ALL) cat /var/log/*`
```

- Sudo can store all input/output from a command that ran within sudo?

But wait, sudo actually does a lot more

Sudo allows a lot of customization, did you for example know that...

- User and group names are matched case insensitive by default?
- You can use globs and regexes in specifying which commands you can run?

```
`ruben ALL = (ALL) cat /var/log/*`
```

- Sudo can store all input/output from a command that ran within sudo?
- The sudoers file can filter rules based on the current hostname?

But wait, sudo actually does a lot more

Sudo allows a lot of customization, did you for example know that...

- User and group names are matched case insensitive by default?
- You can use globs and regexes in specifying which commands you can run?

```
`ruben ALL = (ALL) cat /var/log/*`
```

- Sudo can store all input/output from a command that ran within sudo?
- The sudoers file can filter rules based on the current hostname?
- You can ask sudo to insult you when you typed in an incorrect password?

```
`My pet ferret can type better than you!`
```

But wait, sudo actually does a lot more

Sudo allows a lot of customization, did you for example know that...

- User and group names are matched case insensitive by default?
- You can use globs and regexes in specifying which commands you can run?

```
`ruben ALL = (ALL) cat /var/log/*`
```

- Sudo can store all input/output from a command that ran within sudo?
- The sudoers file can filter rules based on the current hostname?
- You can ask sudo to insult you when you typed in an incorrect password?

```
`My pet ferret can type better than you!`
```

- You can change each and every one of these settings on a per group, per user or even per command basis?

Take this gem for example

Say we have the user ``ferris`` in our system

```
Defaults:FeRrIs !case_insensitive_user  
FeRrIs ALL = (ALL) ALL
```

This rule would only match if checking is done case insensitive, but once it matches we should no longer match case insensitive. So what is ferris allowed to do?

Memory safety is just the start

Re-implementing sudo has a lot more advantages

Memory safety is just the start

Re-implementing sudo has a lot more advantages

- We can get rid of a lot of legacy
 - PAM and NSS have taken away the need for a lot of sudo features

Memory safety is just the start

Re-implementing sudo has a lot more advantages

- We can get rid of a lot of legacy
 - PAM and NSS have taken away the need for a lot of sudo features
- We can leverage the Rust typesystem
 - We can use enums and newtypes to prevent mistakes

Memory safety is just the start

Re-implementing sudo has a lot more advantages

- We can get rid of a lot of legacy
 - PAM and NSS have taken away the need for a lot of sudo features
- We can leverage the Rust typesystem
 - We can use enums and newtypes to prevent mistakes
- We can define new interfaces for modern systems
 - We can take into account those forty years of changing surroundings

Memory safety is just the start

Re-implementing sudo has a lot more advantages

- We can get rid of a lot of legacy
 - PAM and NSS have taken away the need for a lot of sudo features
- We can leverage the Rust typesystem
 - We can use enums and newtypes to prevent mistakes
- We can define new interfaces for modern systems
 - We can take into account those forty years of changing surroundings
- We can implement alternative configuration formats
 - Yes, we have heard about ``doas``!

We can use your help

We will be working on getting the basics working

- You can follow our work at [github](#), and try it out when you feel like trying
- If you use `sudo` in a non-standard way we'd love to know about it
- If you have a non-standard `sudoers` file we'd love to review it

Thanks for listening!

Find us on github

 <https://github.com/memorysafety/sudo-rs>