NETFLIX

High-rate filtering with IPFW

Olivier Cochard-Labbé

NETFLIX

Jan 2019, svn 343631, glebius@ **New pfil(9) KPI together with newborn pfil API and control utility**

“New [KA]PI makes it possible to reconfigure pfil(9) configuration: change order of hooks, rehook filter from one filtering point to a different one, disconnect a hook on output leaving it on input only”

“Another future feature is possiblity to create pfil heads, that provide not an mbuf pointer but just a memory pointer with length. That would allow filtering at very early stages of a packet lifecycle, e.g. when packet has just been received by a NIC and no mbuf was yet allocated.”

NETFLIX

April 2019, svn 346247, gallatin@ **mlx5en: Enable new pfil(9) KPI ethernet filtering hooks**

“This allows efficient filtering at packet ingress on mlx5en.

Note that the packets are filtered (and potentially dropped) *before* the driver has committed to (re)allocating an mbuf for the packet. Dropped packets are treated essentially the same as an error. Nothing is allocated, and the existing buffer is recycled. This allows us to drop malicious packets at close to line rate with very little CPU use.”

NETFLIX

NIC drivers pfil(9) ready

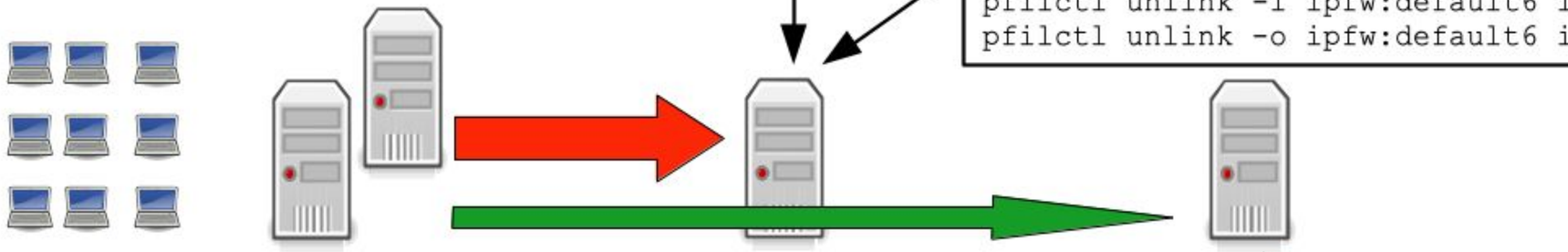
- Mellanox, svn 346247, gallatin@
- vtnet, svn 356613, glebius@
- iflib, svn 346632, gallatin@
- Chelsio, svn 357483, gallatin@

3

NETFLIX

```
###### IPFW in standard mode ######
ipfw table blacklist create type addr
ipfw table blacklist add 198.18.2.0/24
ipfw add deny udp from table\(blacklist\) to any
ipfw add pass ip from any to any
pfilctl unlink -o ipfw:default inet
pfilctl unlink -o ipfw:default6 inet6
```

```
###### IPFW at NIC level mode ######
ipfw table blacklist create type addr
ipfw table blacklist add 198.18.2.0/24
ipfw add deny udp from table\(blacklist\) to any
ipfw add pass ip from any to any
pfilctl link -i ipfw:default-link mce0
pfilctl unlink -i ipfw:default inet
pfilctl unlink -o ipfw:default inet
pfilctl unlink -i ipfw:default6 inet6
pfilctl unlink -o ipfw:default6 inet6
```



packet-generator:
- 20Mpps Legitimate (src: 198.18.1.0/24)
- From 0 to 40Mpps DDoS (src: 198.18.2.0/24)

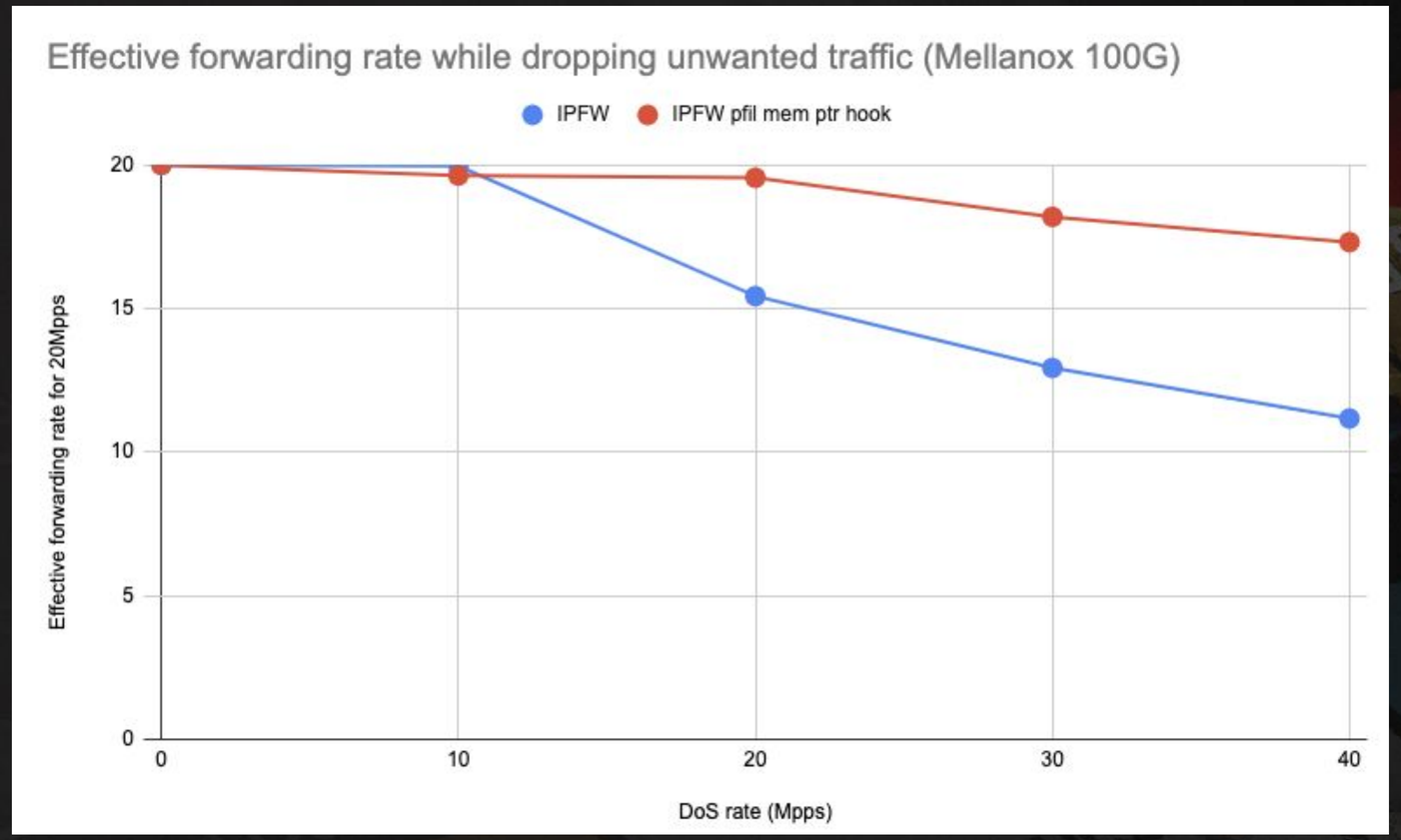
Measuring received packet rate

- Intel Xeon CPU E5-2697A v4 @ 2.60GHz (16 cores, 32 threads)
- Mellanox ConnectX-4 MCX416A-CCAT (QSFP28 100GBASE-SR4)
- Chelsio T580-LP-CR (QSFP+ 40GBASE-SR4)

4

NETFLIX

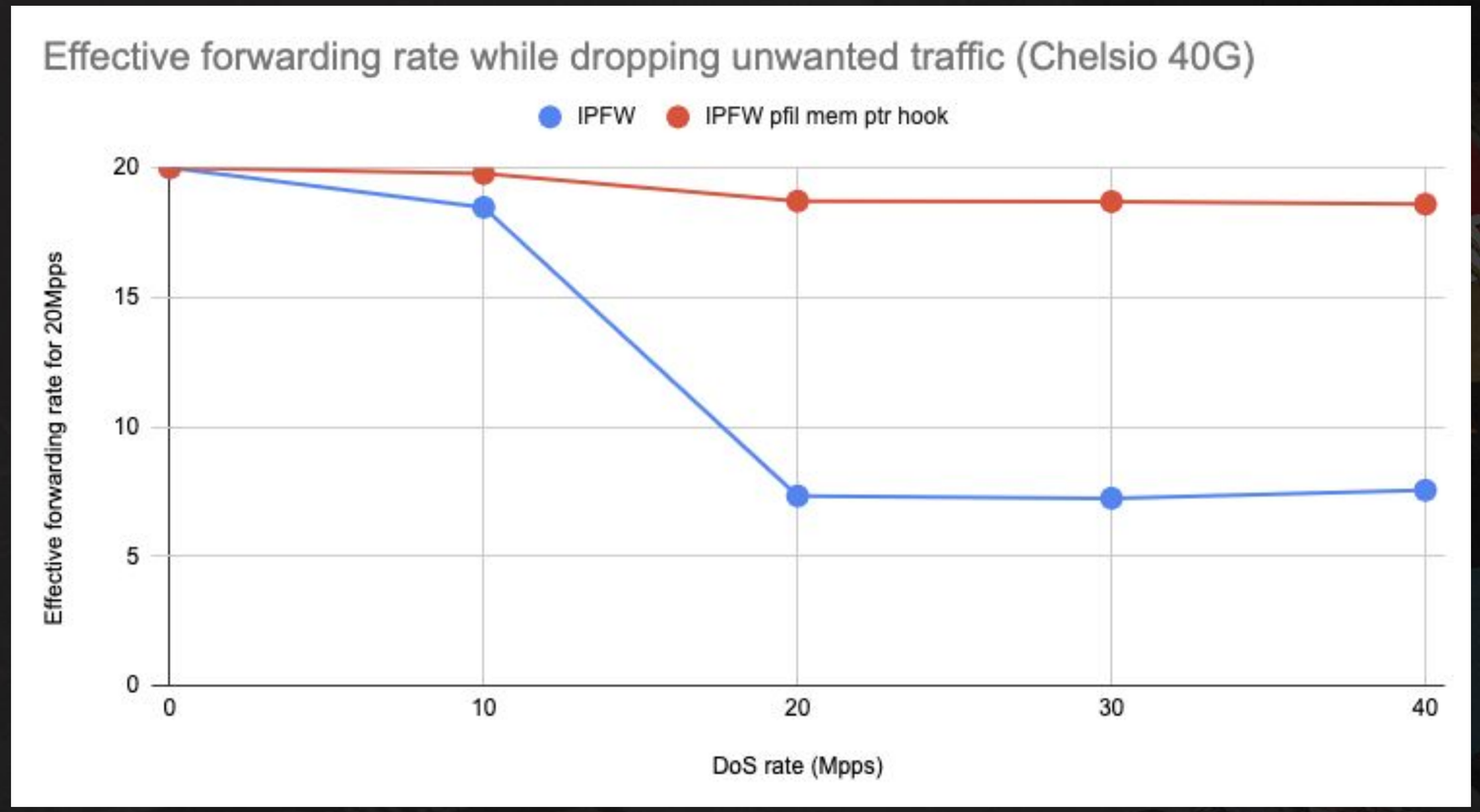
Xeon E5-2697A (16c,32t) and Mellanox 100G



4

NETFLIX

Xeon E5-2697A (16c,32t) and Chelsio 40G



NETFLIX

Results: Legitimate traffic filtered

HW	*NIC*	*Traffic Distribution (legitimate + DoS) in Mpps*	*Legitimate traffic with IPFW in Mpps*	*Legitimate traffic with IPFW + pfil memory pointer in Mpps*	*Improvement*
E5-2697A (16c,32t)	Mellanox 100g	20 + 40	11.17	17.31	53%
	Chelsio 40g	20 + 40	7.56	18.5	146%

HW	*NIC*	*Traffic distribution: Legitimate + DDoS*	*IPFW*	*IPFW at-NIC-level*	*Improvement*
E5-2650L (10c,20t)	Chelsio 10G (8 rxq)	2 Mpps + 12 Mpps	1.14 Mpps	1.92 Mpps	68%

2

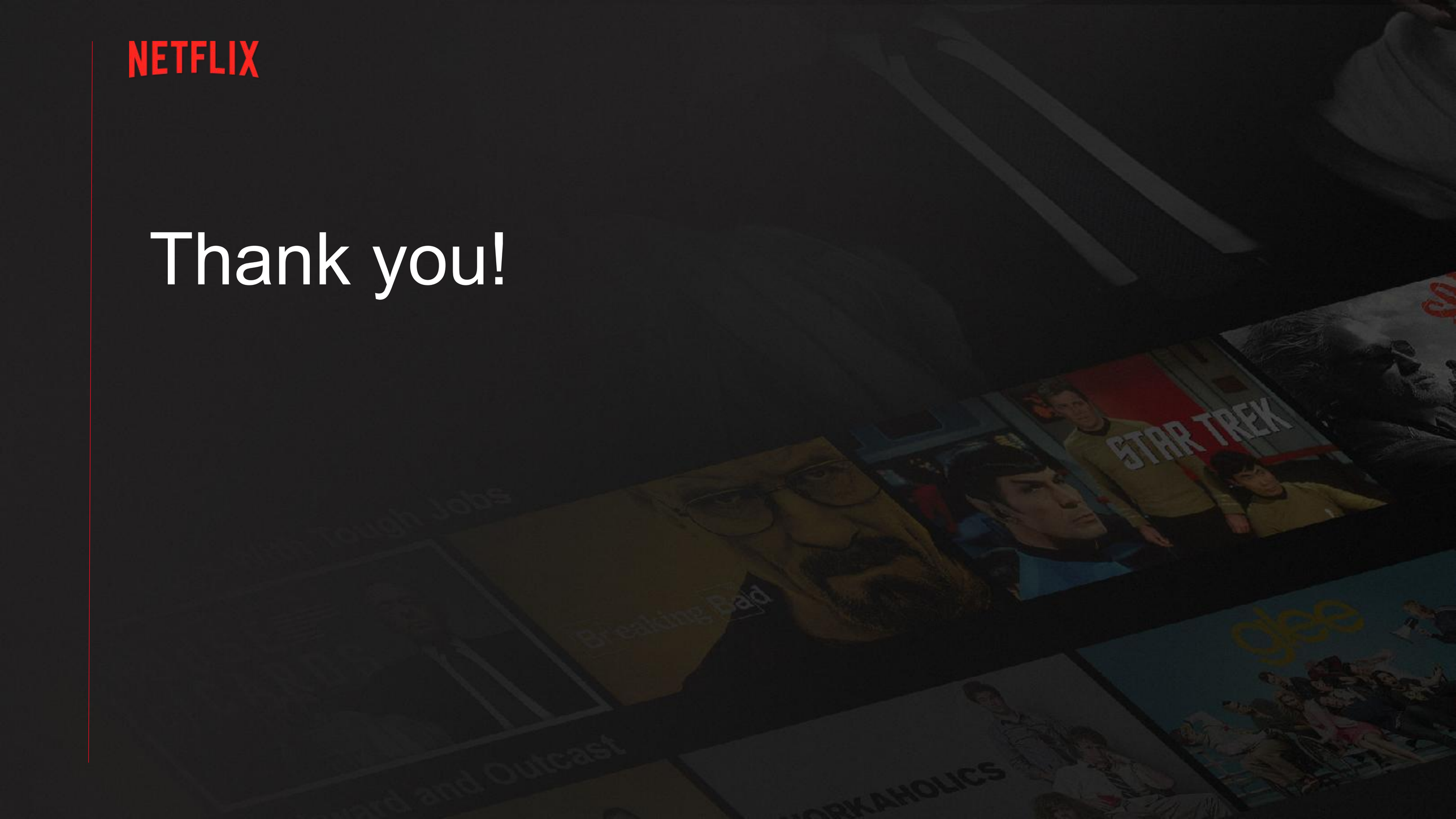
NETFLIX

We are hiring

jobs.netflix.com

NETFLIX

Thank you!



Do you have a Red Pill?

Andrew Cagney
Libreswan

IPsec Interfaces

- FreeBSD
- NetBSD
- OpenBSD

- Just Like a Normal Interface
- Libreswan added BSD support in 5.2

The Blue Pill

west # ip addr show ipsec9

X: ipsec9@NONE: <NOARP,UP,LOWER_UP> mtu 1500 state UNKNOWN

    inet 192.0.1.251/24 scope global ipsec9

west # ip route add 192.0.2.0/24 dev ipsec9

west # ../../guestbin/ping-once.sh --up -I 192.0.1.251 192.0.2.254

up

west # ipsec trafficstatus

#2: "westnet4-eastnet4", type=ESP, add_time=1234567890, inBytes=84, outBytes=84, maxBytes=2^63B, id='@east'

… bring it up

West# ipsec up westnet4-eastnet4

"westnet4-eastnet4" #1: initiating IKEv2 connection to 192.1.2.23 using UDP

"westnet4-eastnet4" #1: sent IKE_SA_INIT request to 192.1.2.23:UDP/500

"westnet4-eastnet4" #2: initiator established Child SA using #1; IPsec tunnel [192.0.1.0/24===192.0.2.0/24] {ESP/ESN=>0xESPESP <0xESPESP xfrm=AES_GCM_16_256-NONE DPD=passive}

West# ../../guestbin/ping-once.sh --up -I 192.0.1.251 192.0.2.254

up

West# ipsec trafficstatus

#2: "westnet4-eastnet4", type=ESP, add_time=1234567890, inBytes=84, outBytes=84, maxBytes=2^63B, id='@east'

… but when I take the Red Pill

```
# move it into the name space
west # ../../guestbin/ip.sh netns add ns
west # ../../guestbin/ip.sh link set ipsec9 netns ns
west # ../../guestbin/ip.sh -n ns link show ipsec9 type xfrm
X: ipsec9@NONE: <NOARP> mtu 1500 qdisc state DOWN qlen 1000
west # ../../guestbin/ip.sh -n ns addr show ipsec9
X: ipsec9@NONE: <NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
west # # add the address and mark it up
west # ../../guestbin/ip.sh -n ns addr add 192.0.1.251/24 dev ipsec9
west # ../../guestbin/ip.sh -n ns link set ipsec9 up
```

… how?

Increasing Consistency in `man(4)`

Alex Ziaee (ziaee@)

NAME

man - display online manual documentation pages

SYNOPSIS

man [**-adho**] [**-t** | **-w**] [**-M** manpath] [**-P** pager] [**-S** mansect]
[**-m** arch[:machine]] [**-p** [eprtv]] [mansect] page | file ...
man **-K** | **-f** | **-k** expression ...

DESCRIPTION

The **man** utility finds and displays online manual documentation pages. If mansect is provided, **man** restricts the search to the specific section of the manual.

The sections of the manual are:

1. FreeBSD General Commands Manual
2. FreeBSD System Calls Manual
3. FreeBSD Library Functions Manual
4. FreeBSD Kernel Interfaces Manual
5. FreeBSD File Formats Manual
6. FreeBSD Games Manual
7. FreeBSD Miscellaneous Information Manual
8. FreeBSD System Manager's Manual
9. FreeBSD Kernel Developer's Manual

:

NAME

open, **openat** - open or create a file for reading, writing or executing

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <fcntl.h>

int
open(const char *path, int flags, ...);

int
openat(int fd, const char *path, int flags, ...);
```

DESCRIPTION

The file name specified by *path* is opened for either execution or reading and/or writing as specified by the argument *flags* and the file descriptor returned to the calling process. The *flags* argument may indicate the file is to be created if it does not exist (by specifying the O_CREAT flag). In this case **open**() and **openat**() require an additional argument *mode_t mode*, and the file is created with mode *mode* as described in chmod(2) and modified by the process' umask value (see umask(2)).

The **openat**() function is equivalent to the **open**() function except in the

:

NAME

err, **verr**, **errc**, **verrc**, **errx**, **verrx**, **warn**, **vwarn**, **warnc**, **vwarnc**, **warnx**, **vwarnx**, **err_set_exit**, **err_set_file** - formatted error messages

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <err.h>

void
err(int eval, const char *fmt, ...);

void
err_set_exit(void (*exitf)(int));

void
err_set_file(void *vfp);

void
errc(int eval, int code, const char *fmt, ...);

void
errx(int eval, const char *fmt, ...);
```

:

NAME

hack - exploring The Dungeons of Doom

SYNOPSIS

hack [**-d** directory] [**-n**] [**-u** playername]
hack [**-d** directory] [**-s**] [**-X**] [playername ...]

DESCRIPTION

hack is a display oriented dungeons & dragons-like game. Both display and command structure resemble rogue. (For a game with the same structure but entirely different display - a real cave instead of dull rectangles - try Quest.)

To get started you really only need to know two commands. The command **?** will give you a list of the available commands and the command **/** will identify the things you see on the screen.

To win the game (as opposed to merely playing to beat other people's high scores) you must locate the Amulet of Yendor which is somewhere below the 20th level of the dungeon and get it out. Nobody has achieved this yet and if somebody does, he will probably go down in history as a hero among heroes.

When the game ends, either by your death, when you quit, or if you escape from the caves, **hack** will give you (a fragment of) the list of top

:

NAME

sysctl - get or set kernel state

SYNOPSIS

sysctl [**-j** jail] [**-bdeFhiJlNnoqTtVWx**] [**-B** bufsize] [**-f** filename]
name[=value[,value]] ...
sysctl [**-j** jail] [**-bdeFhJlNnoqTtVWx**] [**-B** bufsize] **-a**

DESCRIPTION

The **sysctl** utility retrieves kernel state and allows processes with appropriate privilege to set kernel state. The state to be retrieved or set is described using a "Management Information Base" ("MIB") style name, described as a dotted set of components.

The following options are available:

-A Equivalent to **-o -a** (for compatibility).

-a List all the currently available values except for those which are opaque or excluded from listing via the CTLFLAG_SKIP flag. This option is ignored if one or more variable names are specified on the command line.

-B bufsize Set the buffer size to read from the **sysctl** to bufsize. This is necessary for a **sysctl** that has variable length, and

:

NAME

boottime, **time_second**, **time_uptime** - system time variables

SYNOPSIS

```
#include <sys/time.h>

extern struct timeval boottime;
extern time_t time_second;
extern time_t time_uptime;
```

DESCRIPTION

The boottime variable holds the estimated system boot time. This time is initially set when the system boots, either from the RTC, or from a time estimated from the system's root filesystem. When the current system time is set, stepped by ntpd(8), or a new time is read from the RTC as the system resumes, boottime is recomputed as new_time - uptime. The sysctl(8) kern.boottime returns this value.

The time_second variable is the system's “wall time” clock to the second.

The time_uptime variable is the number of seconds since boot.

The bintime(9), getbintime(9), microtime(9), getmicrotime(9), nanotime(9), and getnanotime(9) functions can be used to get the current time more accurately and in an atomic manner. Similarly, the

:

NAME

liquidio - Cavium 10Gb/25Gb Ethernet driver

SYNOPSIS

To compile this driver into the kernel, place the following line in your kernel configuration file:

```
device lio
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
if_lio_load="YES"
```

DESCRIPTION

The **liquidio** driver provides support for 23XX 10Gb/25Gb Ethernet adapters. The driver supports Jumbo Frames, Transmit/Receive checksum offload, TCP segmentation offload (TSO), Large Receive Offload (LRO), VLAN tag insertion/extraction, VLAN checksum offload, VLAN TSO, and Receive Side Steering (RSS)

Support for Jumbo Frames is provided via the interface MTU setting. Selecting an MTU larger than 1500 bytes with the ifconfig(8) utility configures the adapter to receive and transmit Jumbo Frames. The maximum MTU size for Jumbo Frames is 16000.

:

NAME

mtw - MediaTek MT7601U USB IEEE 802.11n wireless network driver

SYNOPSIS

```
device usb
device mtw
device wlan

In rc.conf(5):
kld_list="if_mtw"

In sysctl.conf(5):
hw.usb.mtw.debug=0xffffffff
```

DESCRIPTION

This module provides support for MediaTek MT7601U USB wireless network adapters. If the appropriate hardware is detected, the driver will be automatically loaded with devmatch(8). If driver autoloading is explicitly disabled, enable the module in rc.conf(5). The **mtw** driver can be configured at runtime with ifconfig(8) or at boot with rc.conf(5).

HARDWARE

The **mtw** driver supports MediaTek MT7601U based USB wireless network adapters including (but not all of them tested):

:

NETFLIX

Using production environment for tests

Olivier Cochard-Labbé

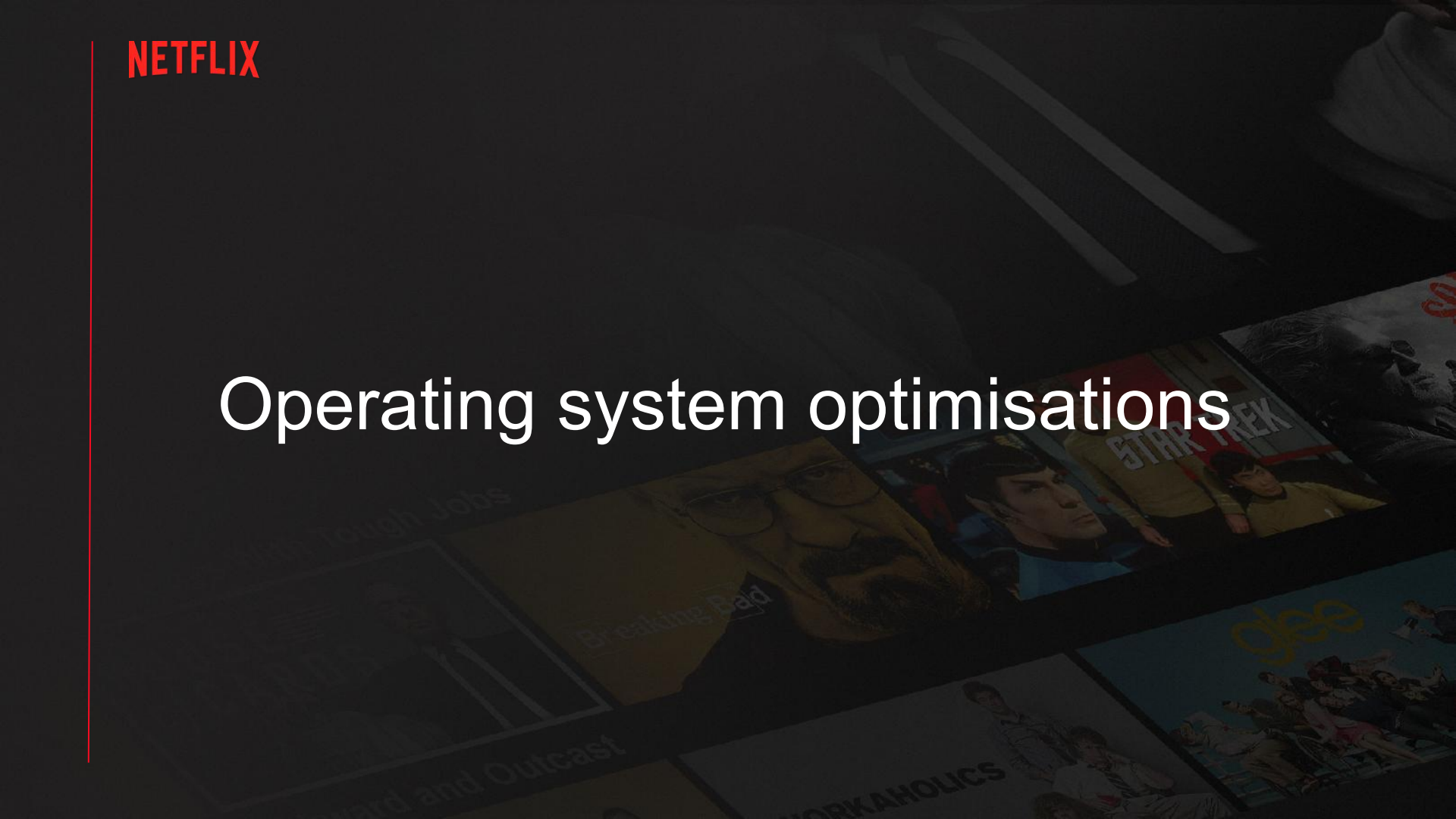
NETFLIX

Definitions

- CDN: Content Delivery Network
- OCA: Open Connect Appliance (CDN server)
- Members (or users)
- Client: Netflix application
- Client devices: smartphone, TV, STB, console, etc.

NETFLIX

Operating system optimisations



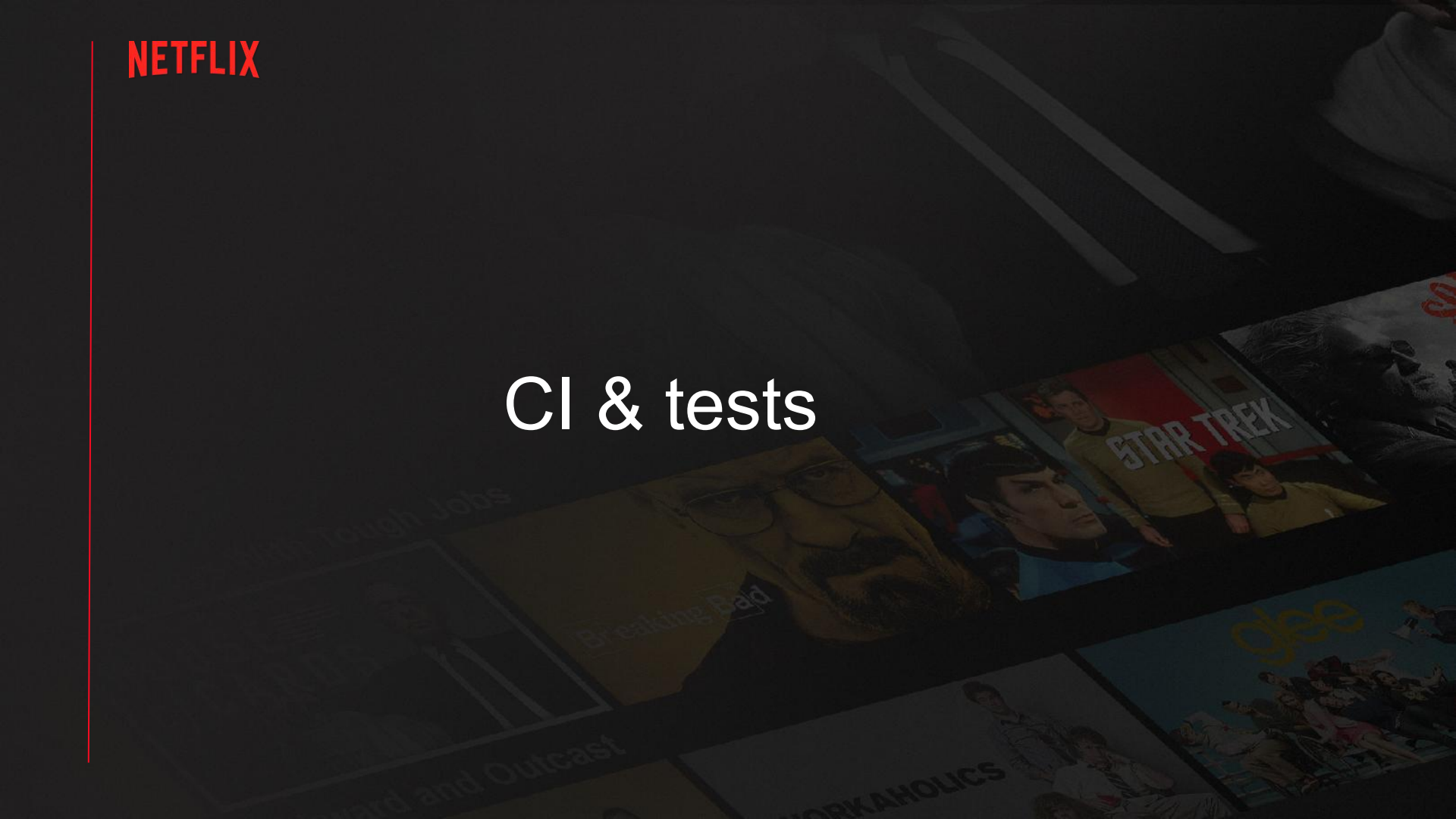
NETFLIX

Serving Stack

- FreeBSD development branch
  - Standard IP stack (no netmap/DPDK)
  - UFS filesystem (ZFS restricted for non-contents disks)
  - Multiples TCP stacks
- NGINX web server
- 100% Open Source (BSD)

NETFLIX

CI & tests



NETFLIX

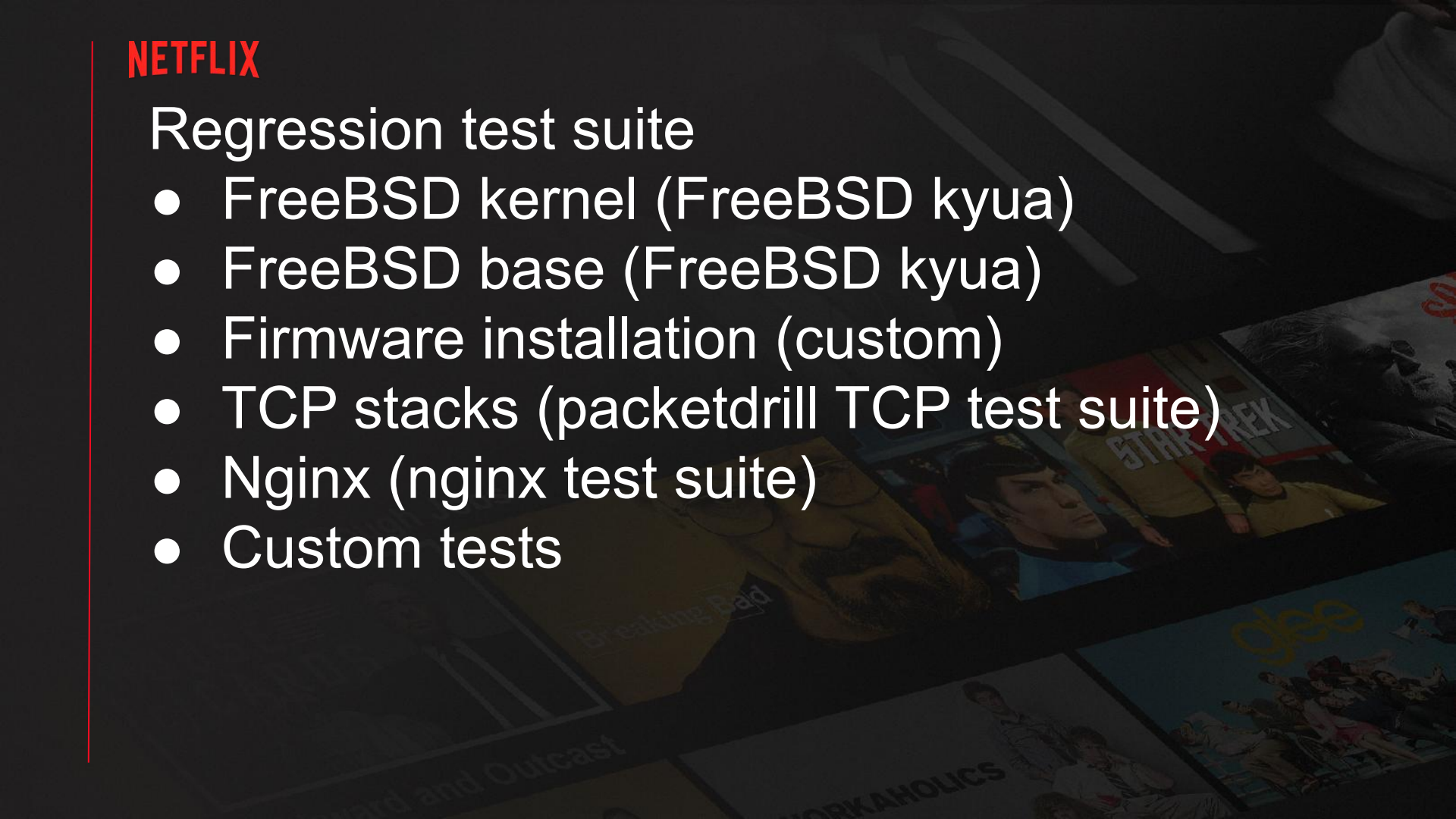
Client devices

- Launched in 2006
- Netflix app in 2010
- Nintendo stopped streaming services Jan 2019
- Still 750K active devices

NETFLIX

Regression test suite

- FreeBSD kernel (FreeBSD kyua)
- FreeBSD base (FreeBSD kyua)
- Firmware installation (custom)
- TCP stacks (packetdrill TCP test suite)
- Nginx (nginx test suite)
- Custom tests



NETFLIX

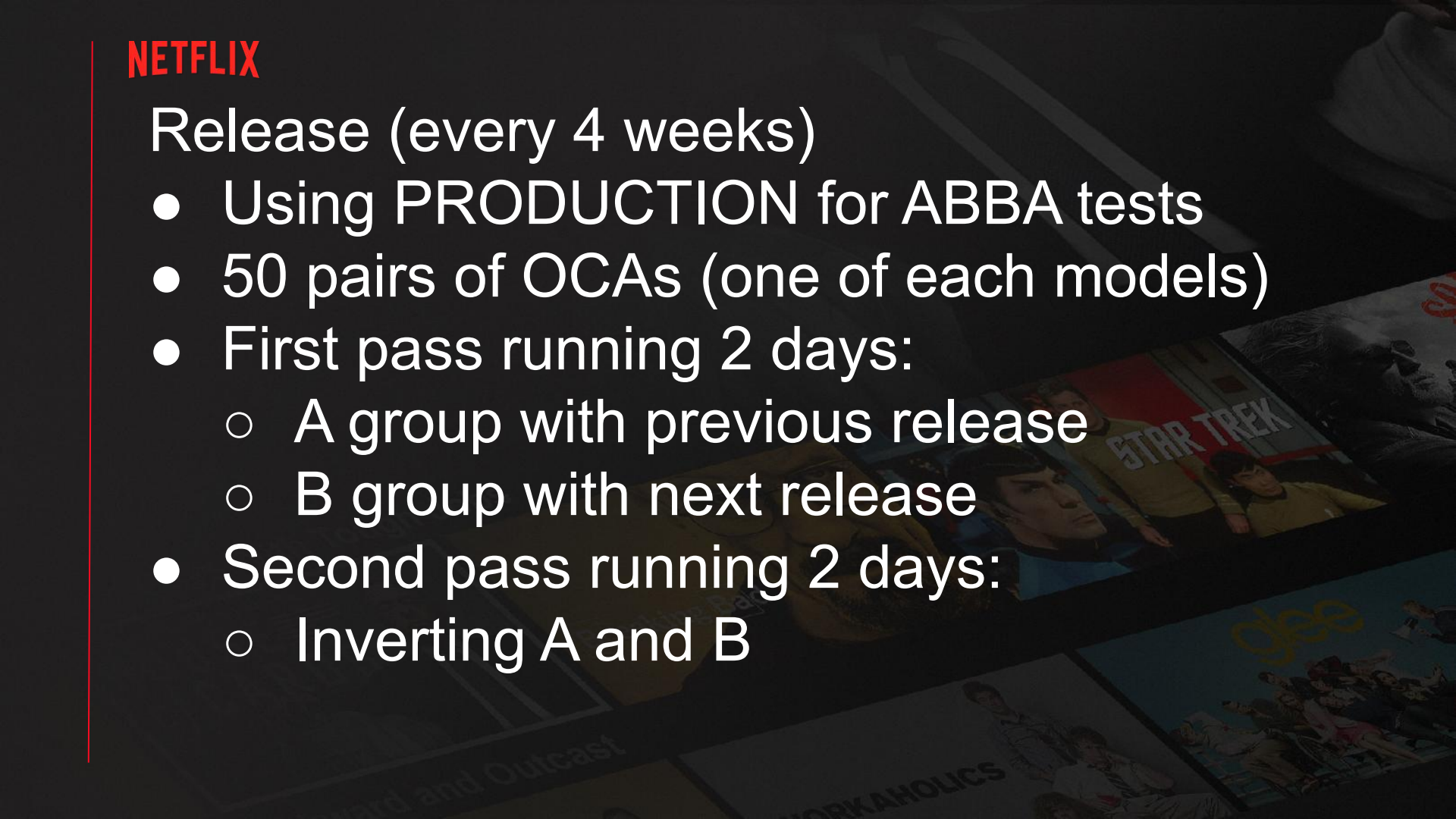
Nightly release (working day)

- Using PRODUCTION for A/B tests
- 50 pairs of OCAs (one of each models) are rebooted:
  - A group with reference release
  - B group with nightly release
- Next morning: check for crash, comparing AB behaviour

NETFLIX

Release (every 4 weeks)

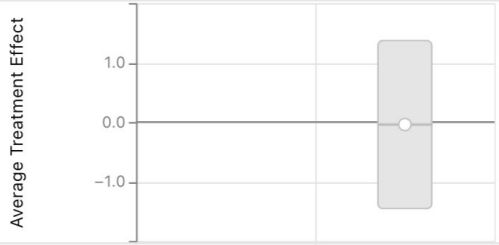
- Using PRODUCTION for ABBA tests
- 50 pairs of OCAs (one of each models)
- First pass running 2 days:
  - A group with previous release
  - B group with next release
- Second pass running 2 days:
  - Inverting A and B



NETFLIX

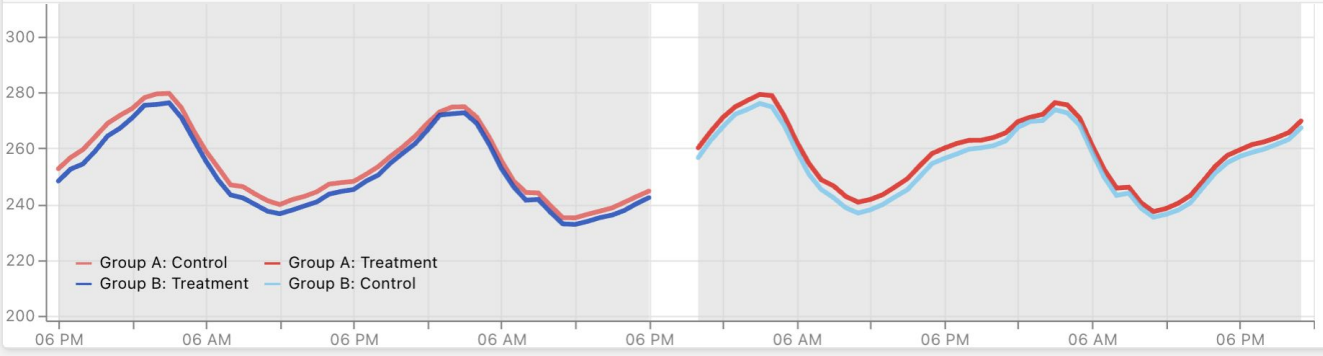
Chassis Power Draw

Firmware Quasi Model



average treatment effect	-	-0.0431
95% confidence interval	-	[-1.46, 1.37]
p-value	-	0.952
sign of treatment effect	-	-1.00

Firmware Quasi Time Series



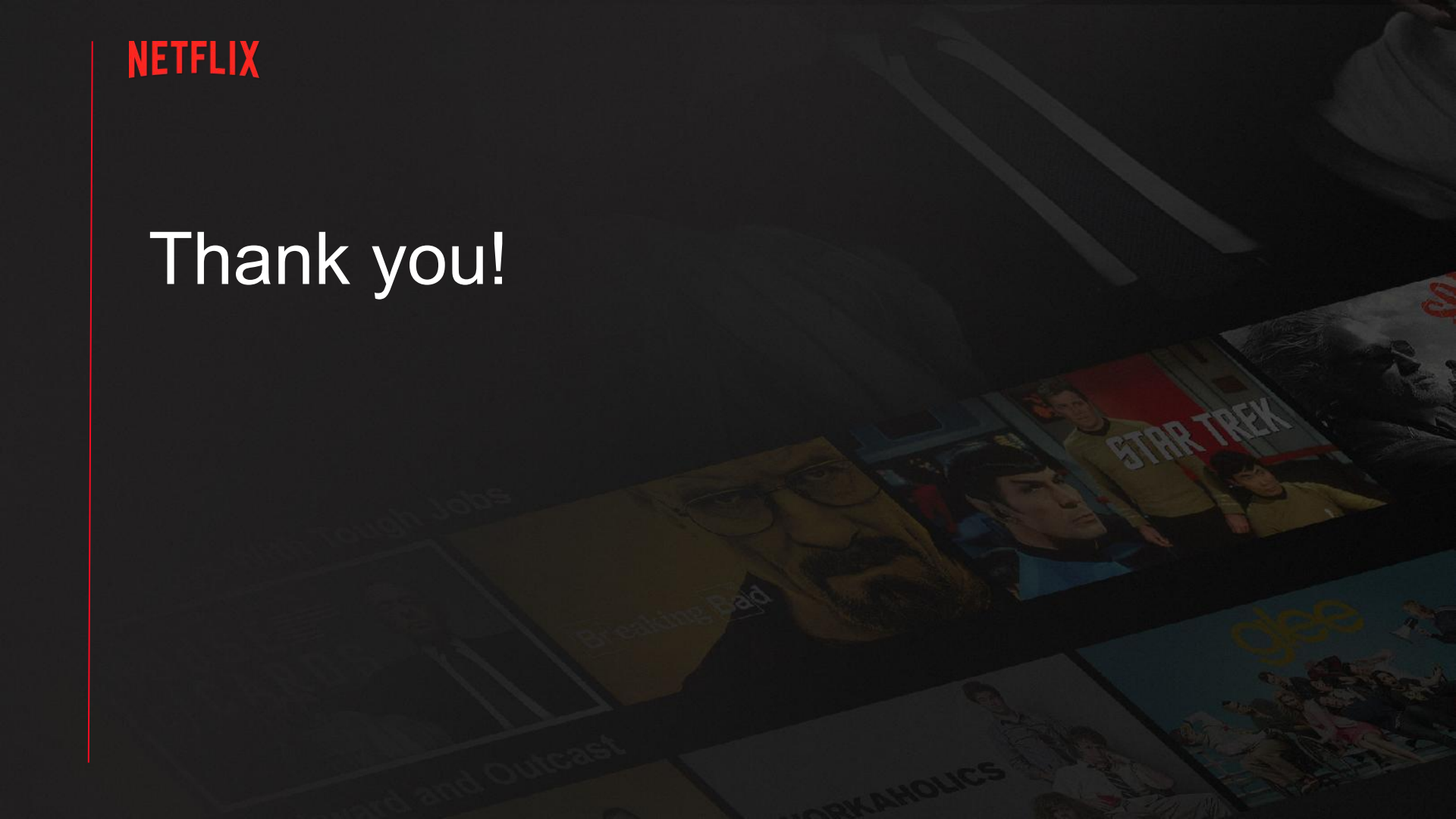
NETFLIX

We are still hiring

jobs.netflix.com

NETFLIX

Thank you!

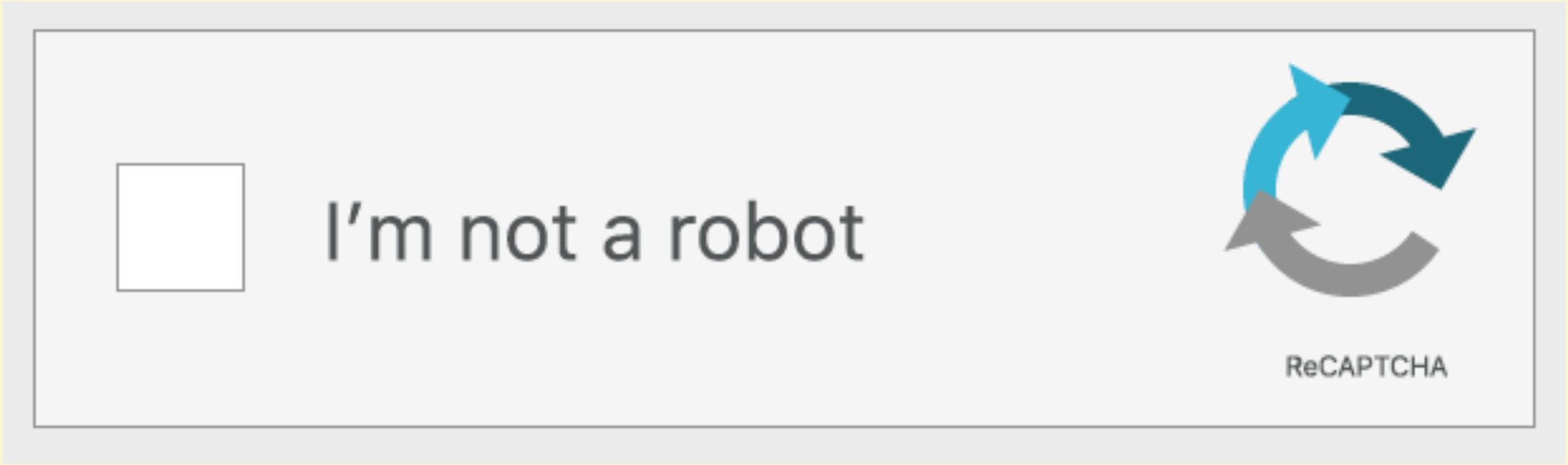


I fight bots in my free time

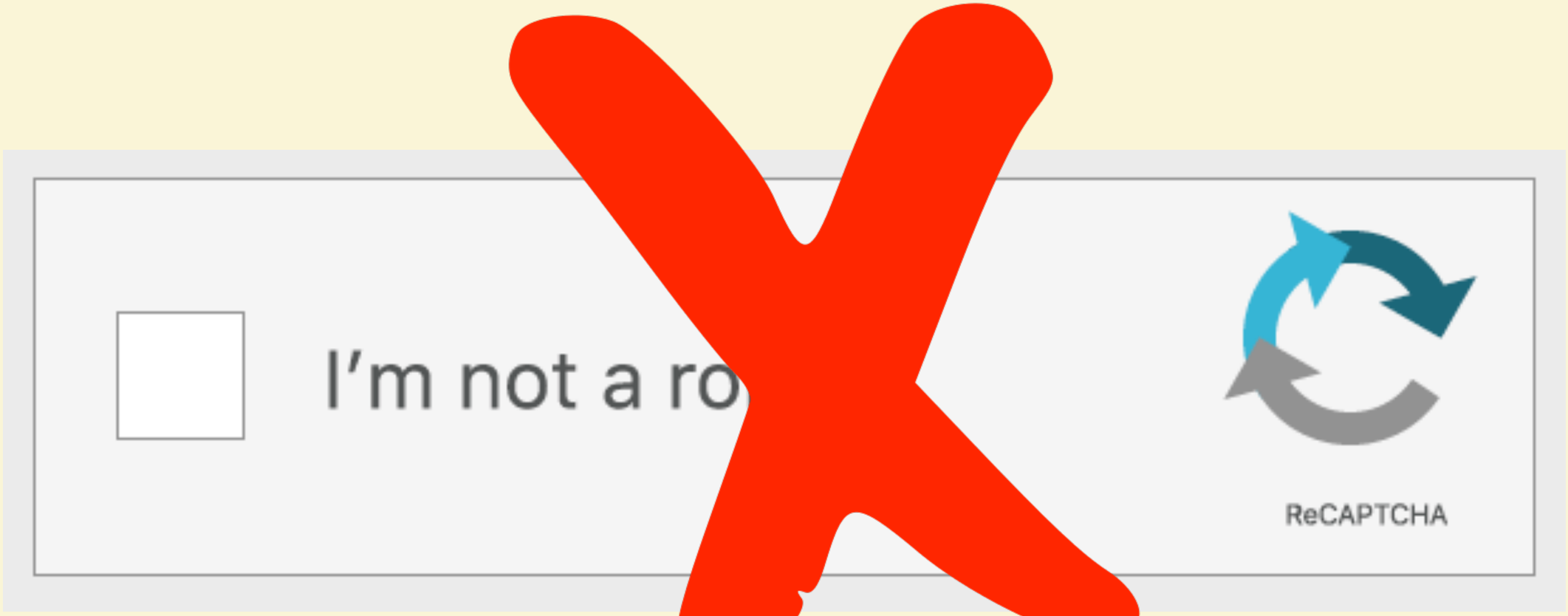
Xe Iaso - CEO @ Techaro

Anubis

Web AI
Firewall Utility

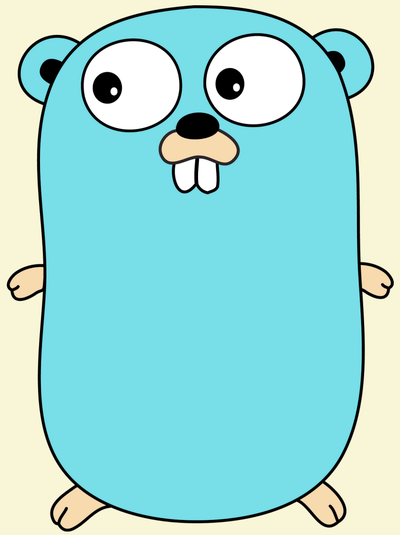

I’m not a robot
ReCAPTCHA

The uncaptcha



Anubis

- Open source software written in Go



TecharoHQ/**anubis**

Weighs the soul of incoming HTTP requests to stop AI crawlers

80 Contributors | 6 Used by | 71 Discussions | 8k Stars | 216 Forks

Anubis

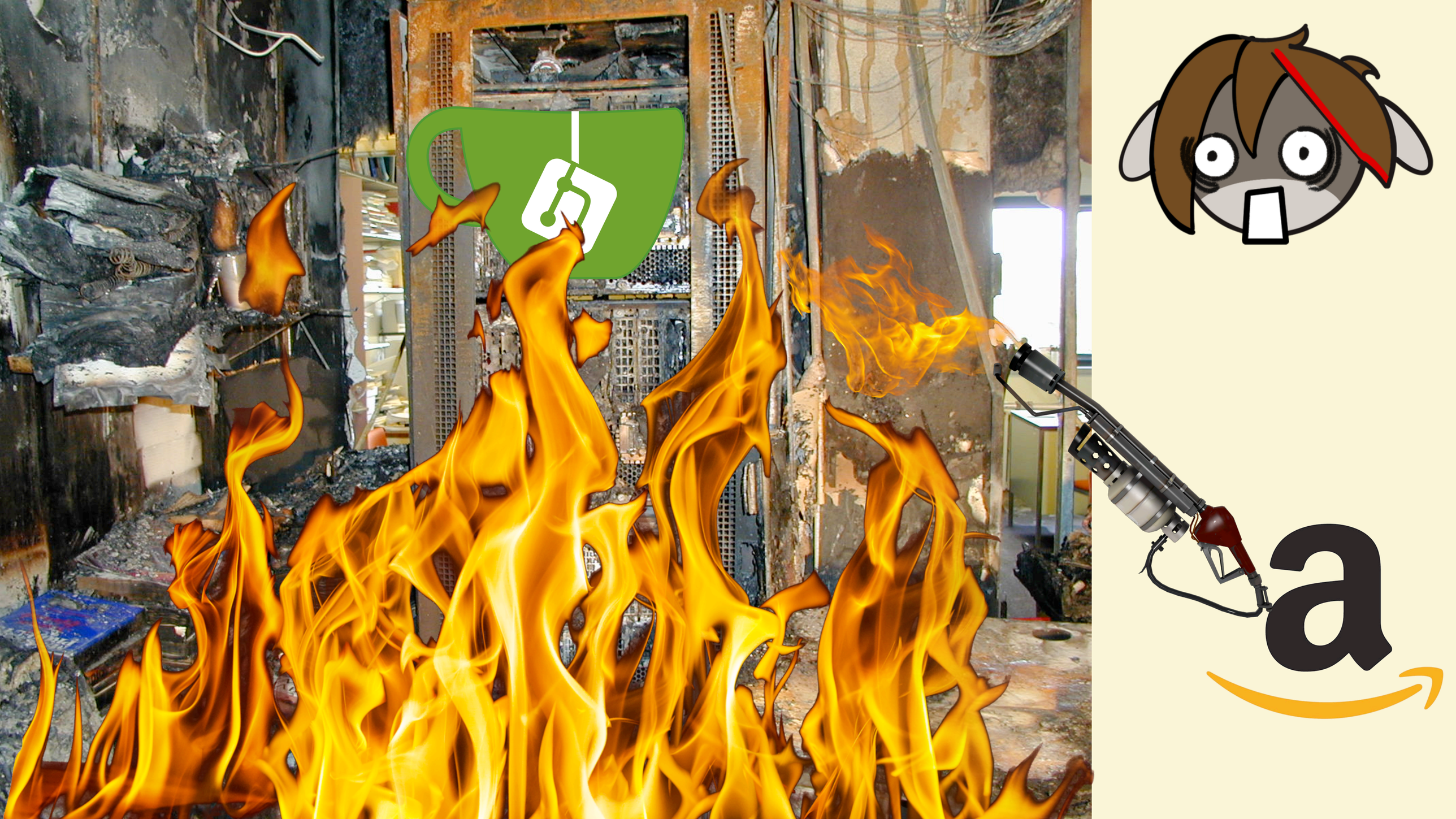
- Open source software written in Go
- Works on any stack that lets you run more than one program

Anubis

- Open source software written in Go
- Works on any stack that lets you run more than one program
- In package repos

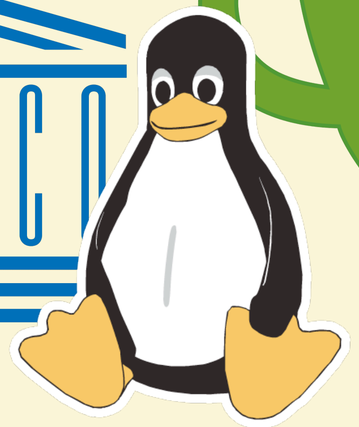
Packaging status					
Alpine Linux Edge	1.19.1	Gentoo overlay GURU	1.19.1	nixpkgs stable 25.05	1.19.1
ALT Linux p11	1.18.0	Homebrew	1.19.1	nixpkgs unstable	1.19.1
ALT Sisyphus	1.18.0	LiGurOS stable	1.17.0	OpenBSD Ports	1.19.1
Arch Linux	1.19.1	LiGurOS develop	1.19.1	Parabola	1.19.1
Arch Linux ARM aarch64	1.19.1	Manjaro Stable	1.18.0	pkgsrc current	1.18.0
AUR	r102.g878b371	Manjaro Testing	1.19.1	Void Linux x86_64	1.18.0
Artix	1.19.1	Manjaro Unstable	1.19.1		
FreeBSD Ports	1.18.0	nixpkgs stable 24.11	1.18.0		

Why does Anubis exist?

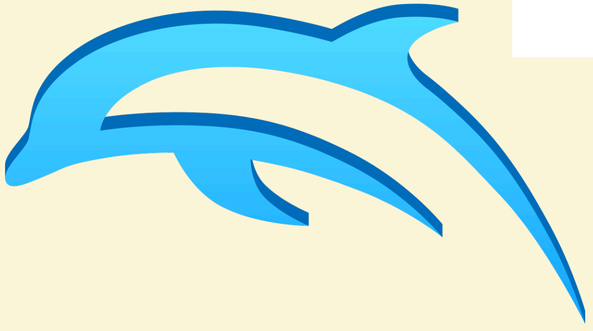


FFMPEG

FreeBSD®

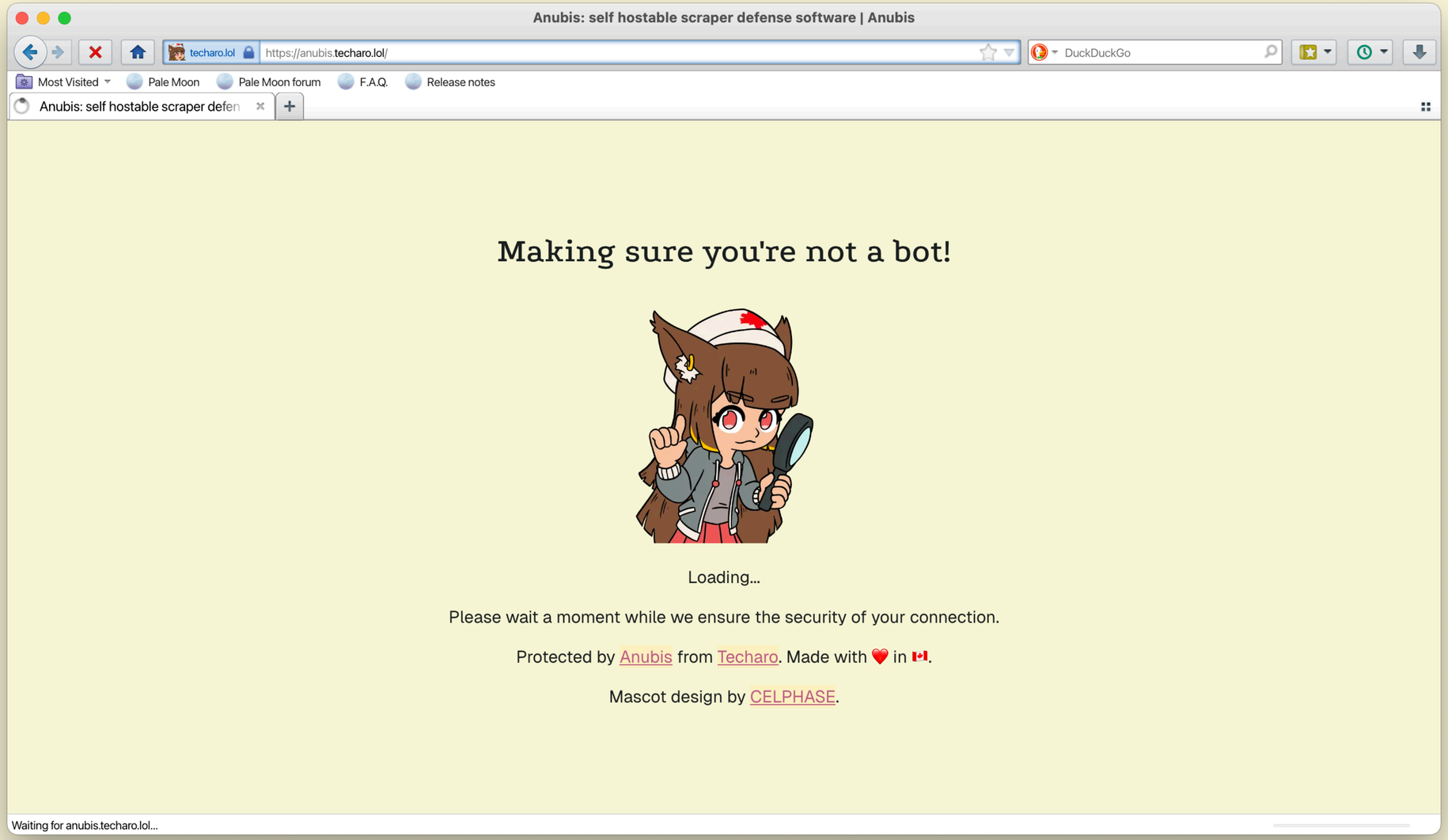


Philipps Universität Marburg



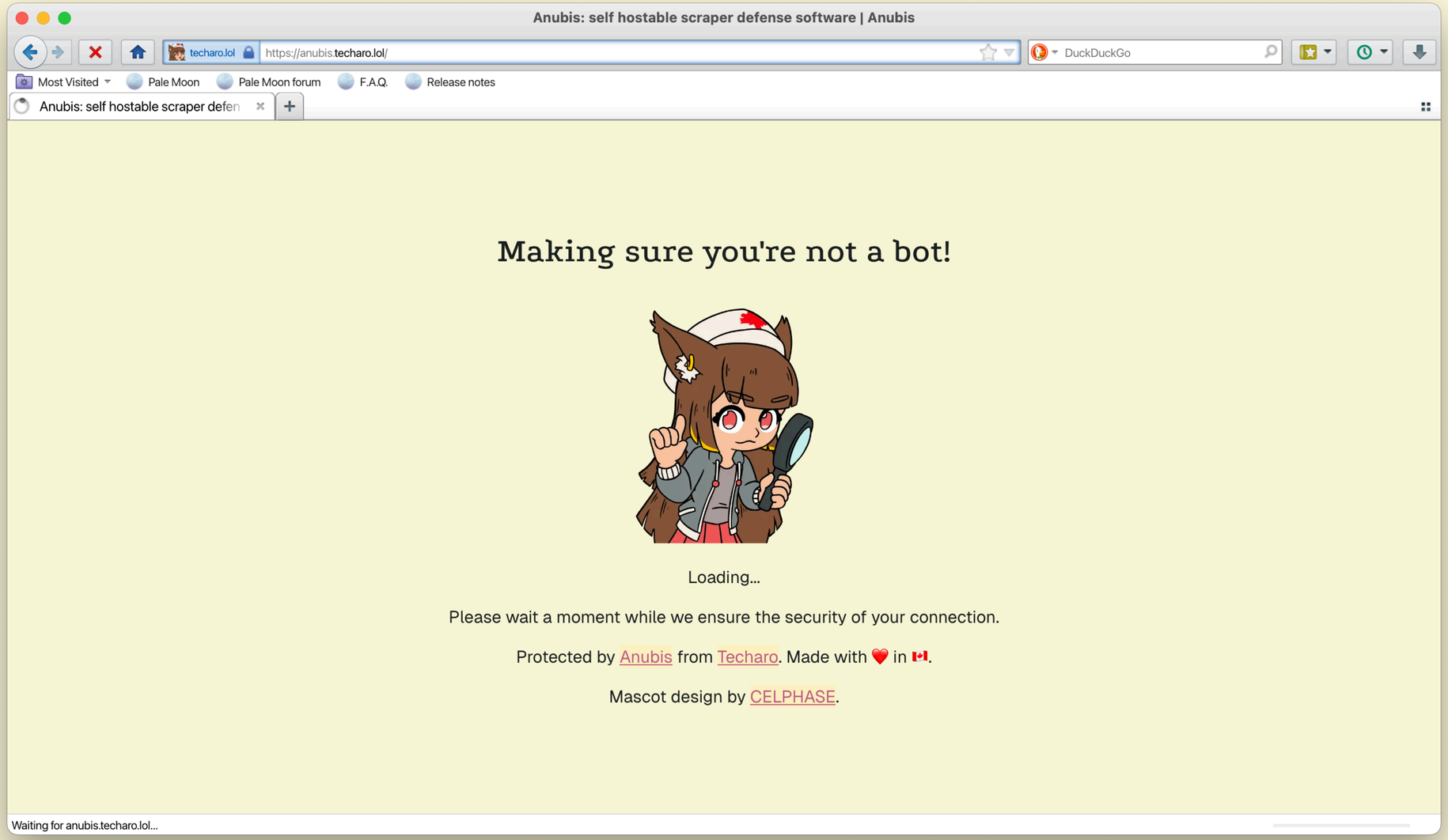
Why is the problem hard?

Is this a browser?



What do scrapers look like?

Sadly, they look like browsers now



Fingerprints I’m trying

- JA4 TLS
- JA3N TLS
- JA4H HTTP (limited success)
- HTTP/2 fingerprinting (limited success)
- THR1 HTTP (my own fingerprint)
- If the client executes JavaScript
  and supports modern JS features

What I want to do next

- Hosted option like CloudFlare
- System load based thresholds
- Better no-JS support
- WebAssembly on the server/client
- IP reputation database (paid, opt-in)
- Kubernetes integration & ingress controller
- Corpo-friendly features
- TLS terminator with fingerprinting for Anubis
- Testing on BSD
- Build binary packages for BSD
- End to end testing that doesn’t suck
- Hire one of the contributors

If you work at an AI company, here's how to sabotage Anubis development

- Quit your job
- Work for Square Enix
- Make absolute banger expansions for Final Fantasy 14

Xe Iaso

Bluesky
@xeiaso.net

TikTok: @xeiaso.1337

https://xeiaso.net

Twitch: @princessxen

I have stickers!

Questions? Concerns?
Intrusive thoughts?

anubis@xeserv.us