

Hello!

Created for EuroBSDCon 2025 in Zagreb, Croatia.

Thanks to the usual crew for support, encouragement,
ideas, critique and – most of all – patience.

You know who you are.

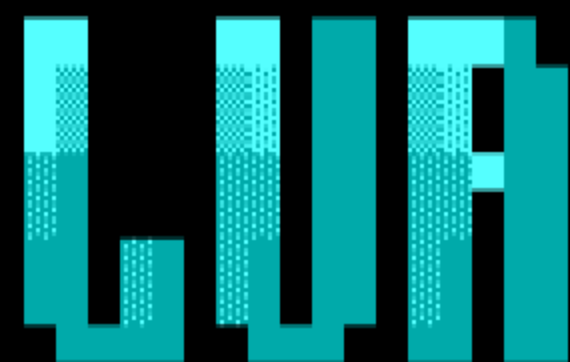


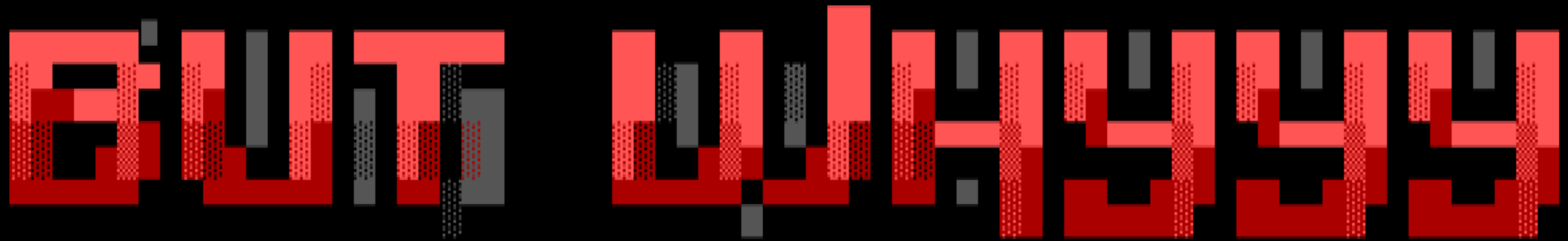
Using `nginx` and `LUA` to thwart
bots and skript kiddies

(Or: Fighting denial-of-service for fun and profit)

A talk about building a Telnet CAPTCHA for your
BBS. And about how you might implement pre-TLS
session management for your website.

At least one of which might belong in the
"why-would-you-even" category..





Because we use Lua to do Real Work(tm)

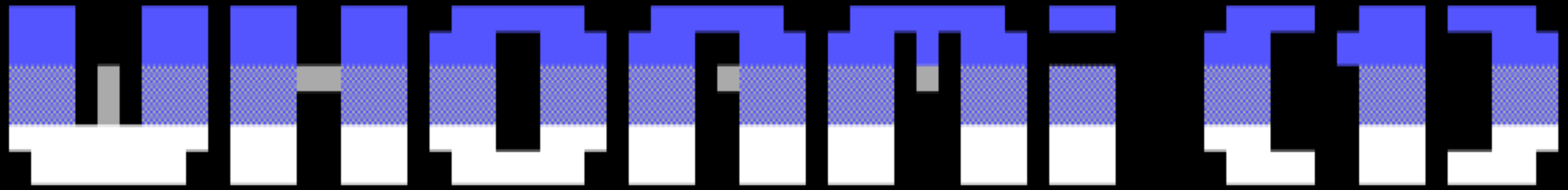
Becuase I'm old enough to be nostalgic...

Because the Internet is a Bad Place(tm)

Could not find any prior art*

Because ... apparently ... I can?

* Ask me during the hallway track..



By day:

- Sysadmin
- FreeBSD
- Payment authentication
- Fighting for open source
- Clinging to the fun parts

By night:

- Retrocomputing nerd
- Hardware hoarder
- Performance artist (making old hardware do silly things)
- Sierra, LucasArts, Beer and MT-32

All the time:

Father of four in an untraditional family. Managed by cats.

CELEBRATING REDISCOVERIES

We used to tell computers what to do

Now computers tell us what we may do

Lifting the hood has become harder

Nobody learns because they have to

...only if they want to

MY DAD AND HIS FRIENDS

Collected and fixed old cars and MCs

Complained about modern vehicles

"Our kids will never understand"

(I did, but only much later)

They used to do it because they had to

ME AND MY FRIENDS

Collect and fix old computers

Complain about modern computers

"Our kids will never understand"

(I can only hope they will .. later)

We used to do it because we had to

a g o o d

midlife crisis

c o s t s

t h e

s a m e

...whether it involves wheels, heels, or ebay deals

..but let's get back on track..

ANDUIN.NET

About

This is `bbs.anduin.net` - Synchronet BBS v2.3
Your sysop: **Ltning** Location: Oslo, Norway

Hardware / Software

CPU: Cx486DRx2-25/50 RAM: 32MB
Modem: Ray Gwinn's SIO/VMODEM
Storage: 8GB CF, NetWare/2 4.12
Operating system: IBM OS/2 v3.0



Safe Space

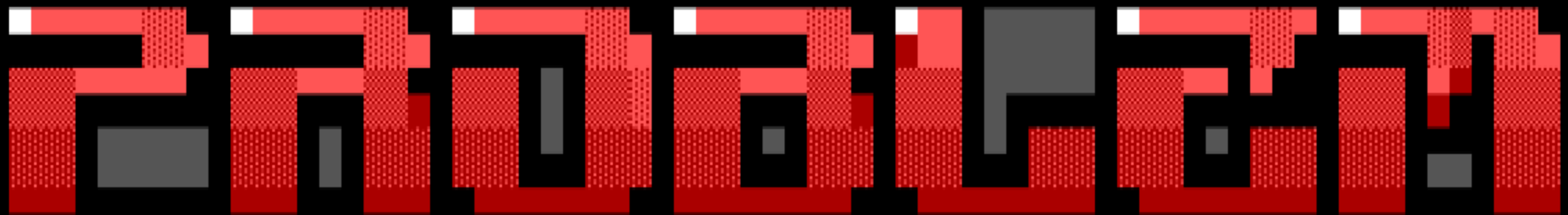
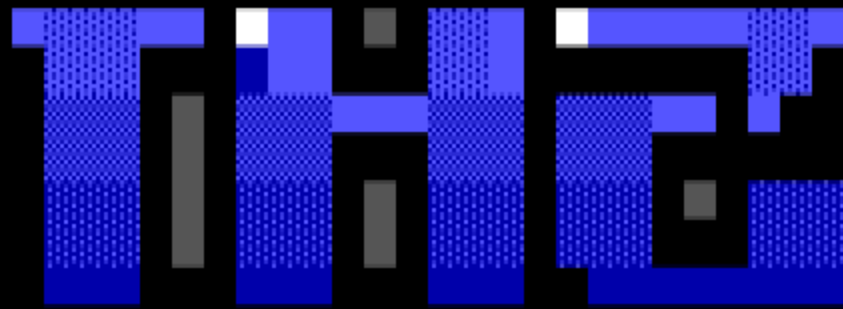
See also WWW:

<http://floppy.museum>

And IRC:

`floppy.museum:6667`

One Ring to rule them all, One Ring to find them, One ring to bring them all
and in the darkness bind them. In the Land of Mordor where the Shadows lie



- A 386-class computer
- Accepts telnet connections
- Emulates a modem for the BBS
- Best case: RING-and-run resource drain
- Worst case: Garbage brings the BBS down

DON'T DISS THE 386!

These old beasts are capable!

(A 286 was linked to from Hacker News – 90k hits in a few hours)

(...I wasn't even around for the show – was giving a talk at NLUUG!)

The 386 is incredibly powerful:

- Virtualization
- "Unlimited" virtual memory
- Actual multi-tasking, thanks to the above

Do not try this at home. Please.

```
stream {
    tcp nodelay on;
    upstream backend {
        server 192.88.99.15:8023 max_conns=2;
    }

    lua_shared_dict lusers 64k;

    server {
        listen 23;
        set $bbs_challenge_passed 0;
        preread_by_lua_file bbs_math.lua;
        log_by_lua_block {
            if tonumber(ngx.var.passed) == 1 then
                local users = ngx.shared.users
                users:incr("active", -1, 1)
            end
        }
        ....
    }
}
```

- Got more boxes? List here
- Tuned to number of nodes on a single box.

- Deceptively simple:
Load challenge from external file
- Abuse log step to maintain counter

```
local sock = assert(ngx.req.socket(true))
math.randomseed(os.time())
local num1 = math.random(1,8)
local num2 = math.random(1,9-num1)
```

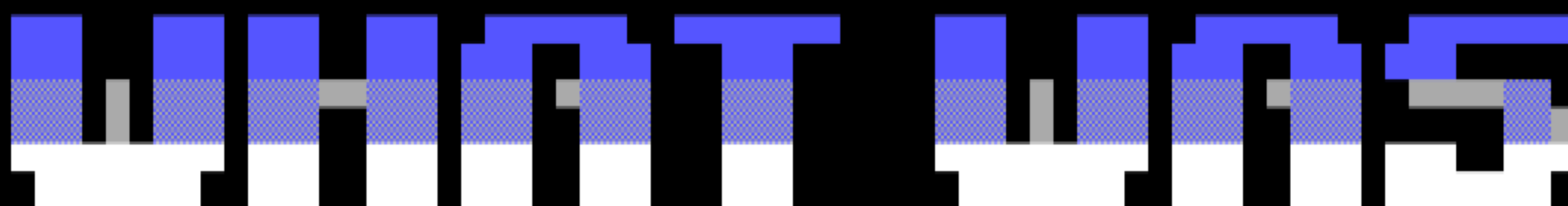
```
ngx.say("\r\n\r\nHello!\r\n\r\n")
ngx.print("What's ", num1, "+", num2, "? ")
```

```
local data = sock:receive(1)
_,_,res = string.find(data, "(%d+)")
```

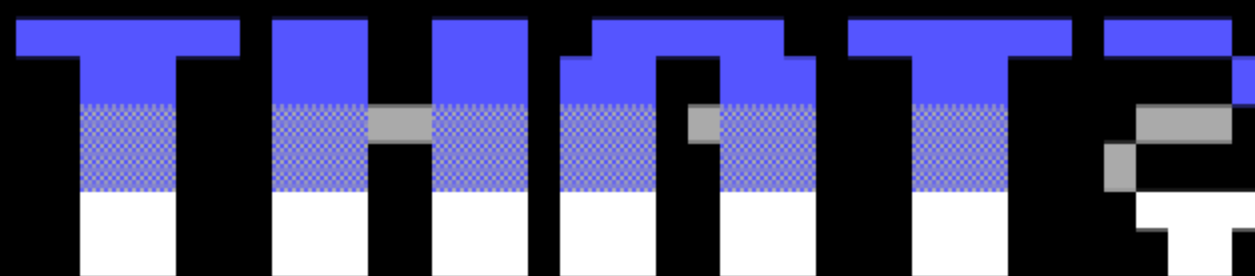
```
if tonumber(res) == num1+num2 then
    ngx.say("\r\n\r\nCorrect!\r\n\r\n")
    ngx.var.bbs_challenge_passed = 1
else
    ngx.say("\r\n\r\nBLEEP!\r\n\r\n")
    ngx.exit(403)
end
```

- Get the socket from ngx
- Don't forget to seed!
- Generate the puzzle..
(not a hard one..)
- Be polite!
- Then be difficult..
- Get a single character
(This is hideously
optimistic! Needs work!)
- Wheee!
- Allow ngx to know..
- Buh-bye...

- Most of the **magic** is in the lua code
- nginx is a good wrapper, doing lots of heavy lifting
- possibly first time it's used for telnet?



- can also be used in anger
- or, as the case may be,
in self-defense



Let's take a closer look!

DEMO

TIME

Demo

TIME

TIMEEEEE!

welcome back

Well that was nice.. But how about something
useful for a change?!?

```
server {
    listen *:81;
    lua_shared_dict sessions 10m;

    location /mysite/ {
        set $C $COOKIE_N;
        access_by_lua_file session_check.lua;
        add_header Set-Cookie "N=$C;Max-Age=86400;";
        proxy_pass http://localhost:1234;
    }

    location / {
        set $C <some random value>;
        header_filter_by_lua_file
            session_create.lua;
        add_header Set-Cookie "N=$C;Max-Age=60;";
        return 302 /mysite/;
    }
}
```

- This is a lot of sessions
- Where your site lives
- Check the session
- Set suitable max age
- Pass to your back-end
- Default landing spot
- May want to preserve if present in request..
- Create the session
- Add the initial header
- Redirect to real site

session_create.lua

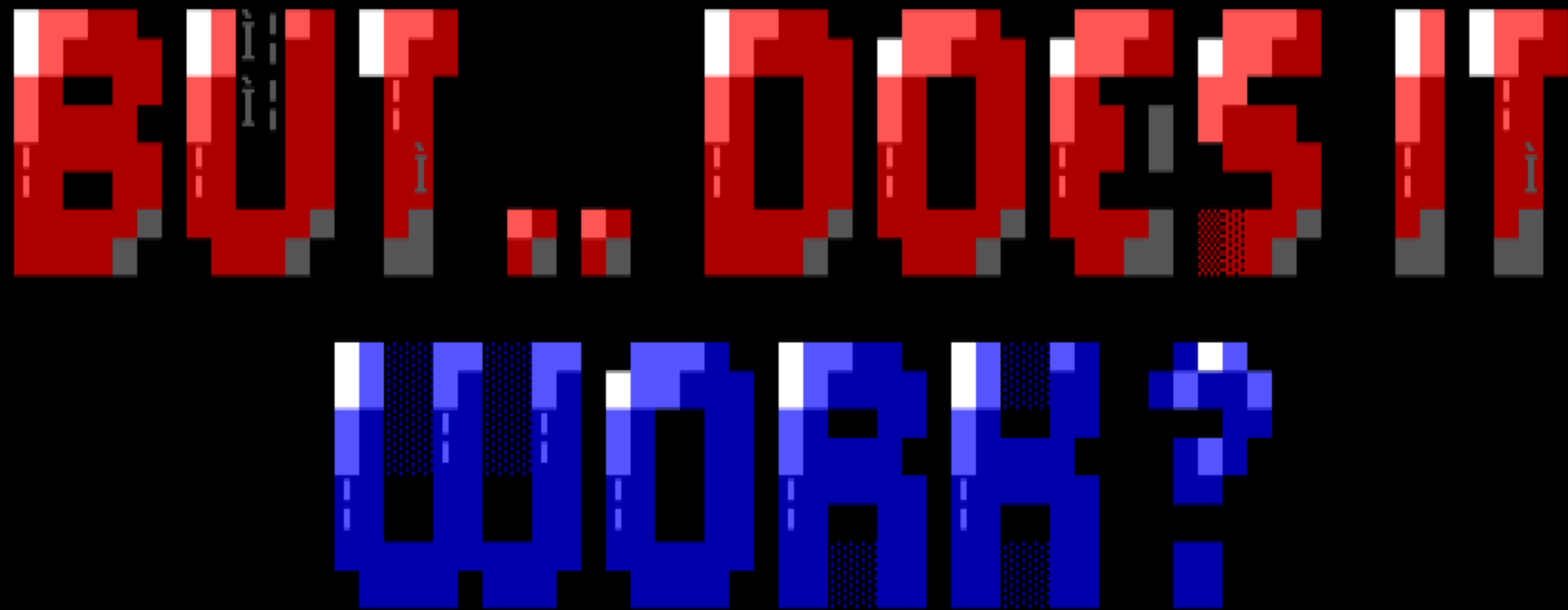
```
local ip = ngx.var.remote_addr
local ua = ngx.var.http_user_agent
local c = ngx.var.C
if (ip and ua and c) then
    local s = ngx.shared.sessions
    s:set(ngx.md5(ip .. ua .. c), "V", 60)
end
```

- Get the data we need for the session ID
- Get the cookie from nginx
- Check if we got all of it
- Pull in the SHM blob
- Add the session ID as key

session_check.lua

```
local ip = ngx.var.remote_addr
local ua = ngx.var.http_user_agent
local c = ngx.var.COOKIE_C
local s = ngx.shared.sessions
local rkey = ngx.md5(ip .. ua .. c)
if not s:get(rkey) then
    ngx.redirect("/")
else
    s:set(rkey, "V", 86400)
end
```

- Get the data
- Get the supplied cookie
- Pull in the SHM blob
- Assemble the session ID
- If not found...
..redirect to /
- If found, bump lifetime



BUT... DOES IT
WORK?

Yes it does – to a point

But it has its limitations.

You may still want something like Anubis

But this works with any client that supports 302 and cookies!

OWTHER IDOAS

Do you want to avoid cookies?

Or maybe save some TLS handshaking?

Stick your session ID in the hostname!

- Set up a wildcard DNS record
- Generate DNS-compliant session IDs
- HTTP 302 to the shiny new host/URL

Bonus points:

- Get your session IDs from elsewhere
- Punish the bots by delaying
- ..or send them to a tarpit

SCRAPERS MUST DIE !!!

```
ssl_client_hello_by_lua_block {  
    local ch =  
        require "ngx.ssl.clienthello"  
    local socket = require 'socket'  
    local s = ngx.shared.sessions  
    local sni, err =  
        ch.get_client_hello_server_name()  
    local host =  
        string.match(sni, '^(.*)')  
    if not s:get(host) then  
        socket.select(nil, nil, 3)  
        ngx.exit(ngx.ERROR)  
    else  
        s:set(host, "V", 86400)  
    end  
}
```



WHY NOT IS IT
GOOD FOR A
a b s o l u t e l y n o t h i n g

- You're still doing a lot of TLS
(although you can do rate limiting during ClientHello)
- That initial connection is still potentially expensive
(but a hell of a lot faster in nginx than in any application)
- We need client-side proof-of-work in TLS
(yes, there is actually standards-body work going on!)

The Talk:

<http://anduin.net/1.htm> ---->

(Also available on a floppy!)

the cast

- in no particular order -

The BBS:

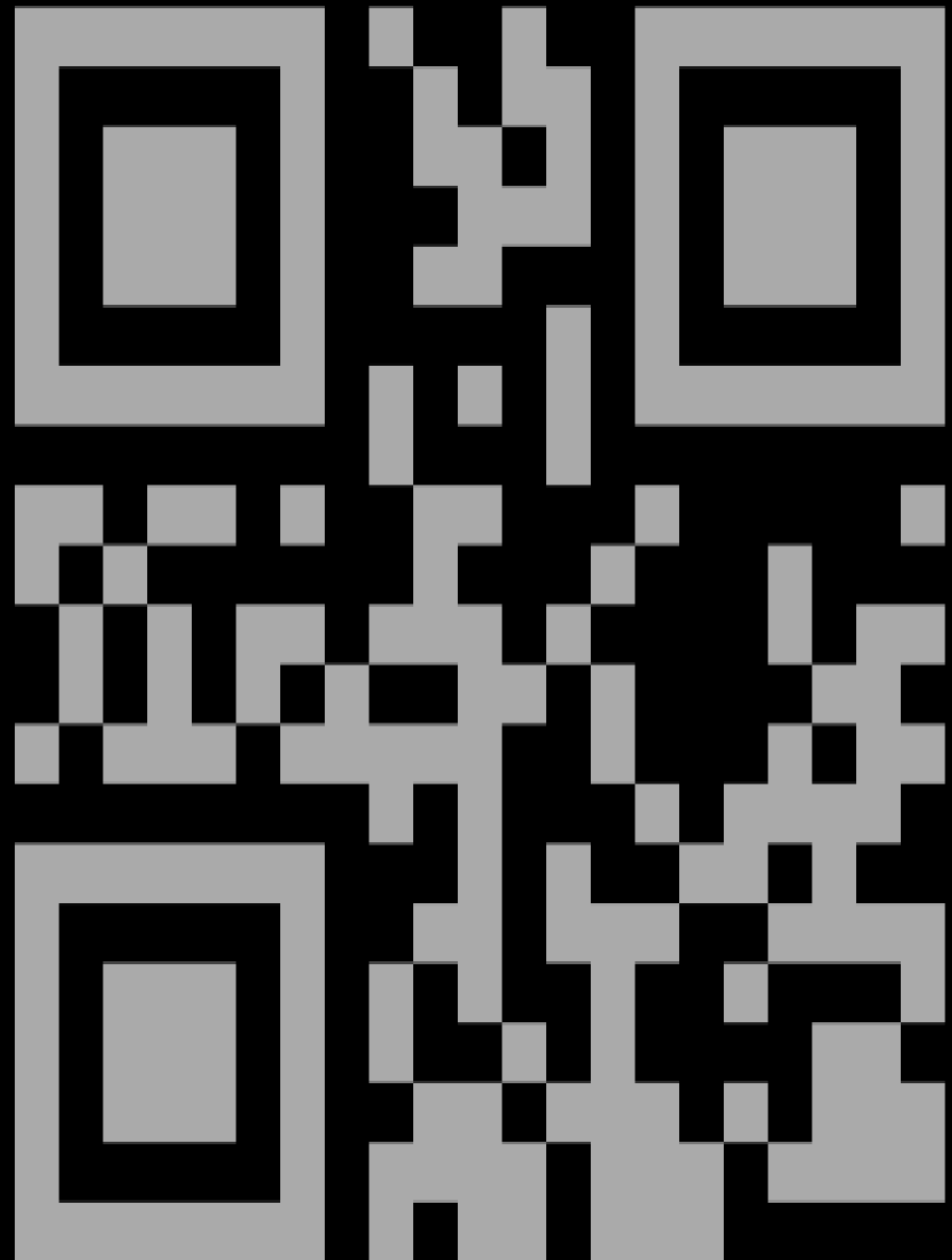
<telnet://bbs.anduin.net:23>

The Chat:

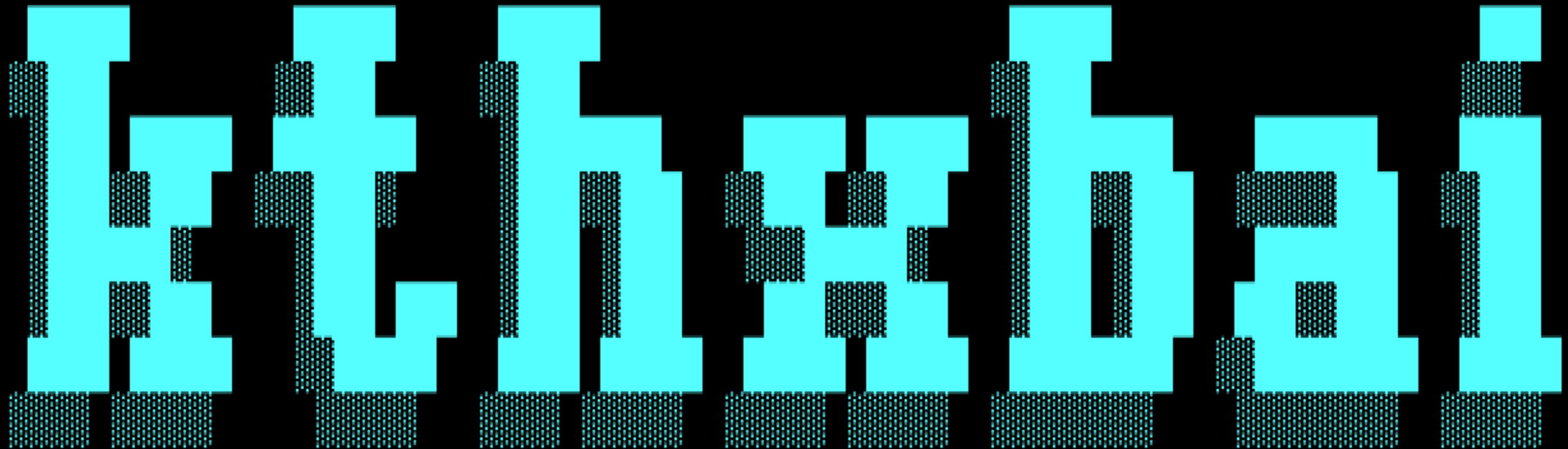
<irc://floppy.museum:6667>

The Floppy Museum:

<http://floppy.museum/>



T-t-that's all, folks!



THANKS

Thank you to the EuroBSDCon organizers,
all our friends in the audience, and the
people who made this (in)sanity possible!

(Don't forget to visit us at our table for free floppies, banter and fun!)