

Obfuscated, self-evaluating bash script by CDN Akamai being supplied to consumers via retail stores

When my wife said to me “Let me show you a t-shirt I saw...”, I wasn’t sure what to expect, but it definitely wasn’t an obfuscated bash script printed on the back designed to print a happy Easter egg message.

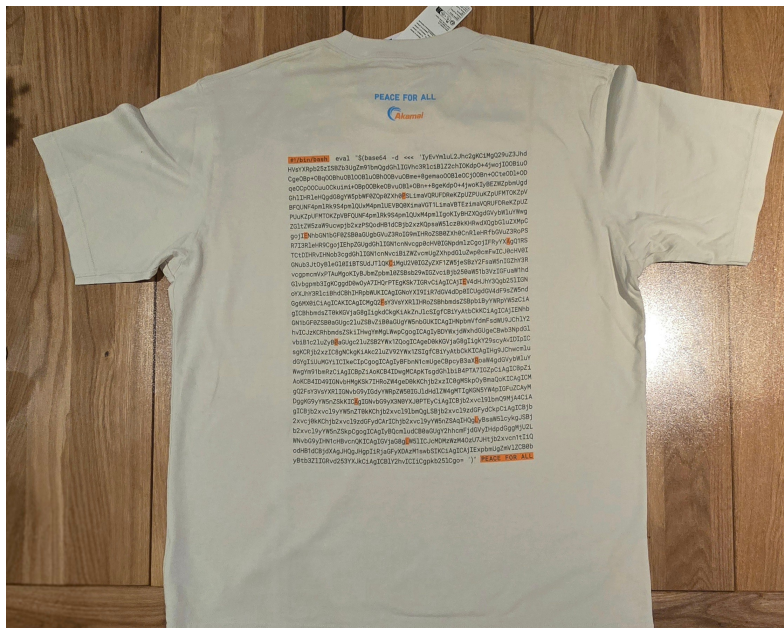
I’M NOT IN THE HABIT of clickbaity headlines¹ but I can see why subeditors have such fun with them. The title above is, strictly speaking, entirely accurate, but probably not what you think. The obfuscated code in question is actually an easter egg, it’s being supplied via Uniqlo stores on an excellent t-shirt designed by Akamai in support of their Peace for All campaign.

And it’s very cool! The front has a heart in curly braces:

¹ I’ve no idea at all how many views this site gets, but I’m willing to bet it’s not even double-digit humans per month.



While the back has a big alphanumeric block:



Is that ... a shebang?!

My wife was right that I'd want to see it. Was that... a shebang?

Take a closer look at the text block:

PEACE FOR ALL



```
#!/bin/bash eval "$(base64 -d <<< 'IyEvYmluL2Jhc2gKCjMgQ29uZ3Jhd
HVsYXRpb25zISBz3UgZm91bmQgdGhlIGVhc3RlciBlZ2chI0KdpO+4jwojI00Biu0
CgeOBp+OBq00Bhu0B100B1u0Bh00Bvu0Bme+8gemao00BleOCj00Bn+OCteOD1+OD
qeOCp0OCuuOCkuimi+OBp00BkeOBvu0B1+OBn++8geKdpO+4jwoKIyBEZWZpbmUgd
Gh1IHRleHQdG8gYW5pbWF0ZQp0ZXh0SLimaVQRUFDRKZpUZPUuKZpUFMTOKZpV
BFQUNF4pm1Rk9S4pm1QUxM4pm1UEVBQ0XimaVGT1LimaVBTEzimaVQRUFDRKZpUZ
PUuKZpUFMTOKZpVBFQUNF4pm1Rk9S4pm1QUxM4pm1IgoKIyBHZXQgdGVybWluYWwg
ZGltZW5zaW9ucwpjb2xzPSQodHB1dCBjb2xzKQpsaW51cz0kKHRwdXQgbGluZXMpC
gojIENhbGN1bGF0ZSB0aGUgbGVuZ3RoIG9mIHRoZSB0ZXh0CnRleHRfbGVuZ3RoPS
R7I3RleHR9CgojIEhpZGUgdGh1IGN1cnNvcgp0cHV0IGNpdmlzCgojIFRyYXAgQ1RS
TCtDIHRvIHNoB3cgdGh1IGN1cnNvciBiZWZvcmUgZXhpdGluZWp0cmFwICJ0cHV0I
GNub3JtOyBleG10IiBTSUdJTLQKQIMgU2V0IGZyZXF1ZW5jeSBzY2FsaW5nIGZhY3R
vcgpmcVxPTAuMgokIyBjb2p0bW0ZSBsb29wIGZvciBjb250aW51b3VzIGFuaW1hd
GlvbGpmb3IgKCGgdD0wOyA7IHQrPTEgKSk7IGRvCiAgICAgIeV4dHJhY3Qgb251IGN
oYXJhY3RlciBhdCBhIHRpbWUKICAgIGNoYXl9IiR7dGV4dDp0ICUgdGV4dF9sZW5nd
Gg6MX0iCiAgICAgICAgICMgQ2sY3VsYXRlIHRoZSBhbmdsZSBpbiByYWRpYW5zCiA
gICBhbmdsZT0kKGVjaG8gIigkdCkgKiAkZnJlcSIgfCBiYyAtbCkKICAgICAgICAgIENhb
GN1bGF0ZSB0aGUgc2luZSBvZiB0aGUgYW5nbGUKICAgIHNpbmVfdmFsdWU9JChlY2
hvICJzKCRhbmdsZSkIHWgYmMgLVWpCgogICAgIyBDYXxjdWxhdGUeCBwb3NpdG1
vbiB1c2luZyB0aGUgc2luZSB2YWx1ZQogICAgID0kKGVjaG8gIigkY29scyAvIDIpIC
sgKCRjb2xzIC8gNCKgKiAk2luZV92YWx1ZSIgfCBiYyAtbCkKICAgIHg9JChwcm1u
dGYgIiUuMGYiICikeCIPcGogICAgIyBFbnN1cmUgeCBpcyB3aXRoZW4gdGVybWluY
WwgYm91bmRzCiAgICBpZiAoKCB4IDwgMCApKTSgdGh1biB4PTA7IGZpCiAgICBpZi
AoKCB4ID49IGNvbHMgKSk7IHRoZW4geD0kKChjb2xzIC0gMSkpOyBmaQoKICAgICM
gQ2FsY3VsYXRlIHNvbG9yIGdyYWRpZW50IGJldHdlZW4gMTIgKGN5YW4pIGFuZCAyM
DggKGYyYW5nZSkKICAgIGNvbG9yX3N0YXJ0PTEyCiAgICBjb2xvc191bmQ9MjA4CiA
gICBjb2xvc19yYW5nZT0kKChjb2xvc191bmQgLSBjb2xvc19zdGFydCkpcyAgICAgICBjb
2xvcj0kKChjb2xvc19zdGFydCArIChjb2xvc19yYW5nZSAqIHQgLyBsaW51cykgJSBj
b2xvc19yYW5nZSkpCgogICAgIyBQcm1udCB0aGUy2hhcmFjdGVyIHdpdGggMjU2L
WNvbG9yIHN1cHBvcnQKICAgIGVjaG8gW51ICJcMDMzWzM4OzU7JHtjb2xvcn1tIiQ
odHB1dCBjdXAgJHQgJHJpIiRjaGFyXDAzM1swbSIKCiAgICAgIExpbmUgZmVlZCB0b
yBtb3ZlIGRvd253YXJkCiAgICBlY2hvICiCgpkb251Cgo= ')" PEACE FOR ALL
```

Yes, a shebang! On a t-shirt sold in a high street store, no less. And it is clearly a base64 encoded Here string being fed to eval via base64 --decode.

Interesting. I told my wife "that's basically how people ship viruses' and bought it.

OCR was cumbersome

There was good news and bad news:

- The bad news was that base64 hasn't got error correction, meaning that the

Sigh.

- The good news was that the string seemed to be intact – at least, it terminated with the expected padding and had matching quotes and braces. This is a good thing because Uniqlo x Akamai sells another design of shirt in the same range which is plainly *incomplete*², a truncated crop from a wider text block which could never compile.

I ran OCR in a few ways: First, using the built-in OCR of the circle-to-search feature on Android, which is often very good. Second, by using Tesseract with a few options and tweaks. And third by running it through Claude. After diffing the three to look for mismatches and getting Claude to output a table of locations for quick scanning, it became trivial but time-consuming to tidy up the remainder. The resulting string was:

IyEvYmluL2Jhc2gKCjMgQ29uZ3JhdHVzYXRpb25zISBZb3I

The decoded script

After Base64 decoding, the resulting script is a welcoming and nicely commented Easter egg:

```
#!/bin/bash
```

```
# Congratulations! You found the easter egg! 🍷
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□♥

```
# Define the text to animate
```

```
text="♥PEACE♥FOR♥ALL♥PEACE♥FOR♥ALL♥PEACE♥FOR♥AI"
```

```
# Get terminal dimensions
```

² For example, its imports are truncated and it ends “retu” instead of “return”. This is a pity, because it’s a really nice colour combination and contains the highly idiomatic instruction `go doStuff(msg, work...` which anyone can relate to.

```

cols=$(tput cols)
lines=$(tput lines)

# Calculate the length of the text
text_length=${#text}

# Hide the cursor
tput civis

# Trap CTRL+C to show the cursor before exiting
trap "tput cnorm; exit" SIGINT

# Set frequency scaling factor
freq=0.2

# Infinite loop for continuous animation
for (( t=0; ; t+=1 )); do
    # Extract one character at a time
    char="${text:t % text_length:1}"

    # Calculate the angle in radians
    angle=$(echo "($t) * $freq" | bc -l)

    # Calculate the sine of the angle
    sine_value=$(echo "s($angle)" | bc -l)

    # Calculate x position using the sine value
    x=$(echo "($cols / 2) + ($cols / 4) * $sine_value" | bc -l)
    x=$(printf "%.0f" "$x")

    # Ensure x is within terminal bounds
    if (( x < 0 )); then x=0; fi
    if (( x >= cols )); then x=$((cols - 1)); fi

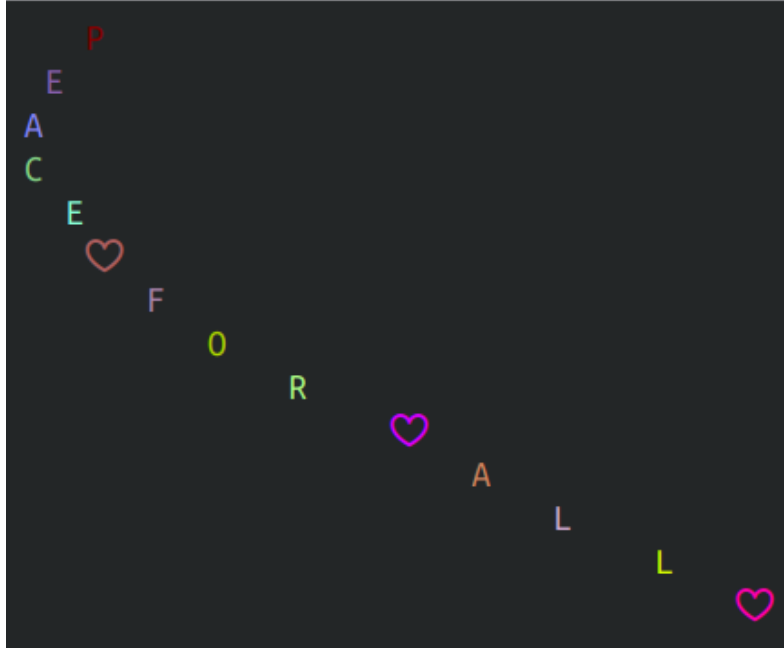
    # Calculate color gradient between 12 (cyan) and 208 (red)
    color_start=12
    color_end=208
    color_range=$((color_end - color_start))
    color=$((color_start + (color_range * t / text_length))

    # Print the character with 256-color support
    echo -ne "\033[38;5;${color}m"${tput cup $x $lines}

    # Line feed to move downward
    echo ""
done

```

The result is a continuous happy sine-wave
loop of the campaign message, Peace for All:



Detail: The font choice

Edit: The following font ID is wrong! User [ralphinus](#) on Hacker News pointed out that the font is Roboto Mono. I don't know how I overlooked the very different g.

I guess Uniqlo is run through Windows though: one thing that struck me was the font, which ~~I'm~~ *I was previously* almost certain is Consolas³, which I'm fond of. Note the very shallowly-slashed 0, the lack of serif on the 1 and the rounded curves of letters like bDyg and number 2. It's striking because it's primarily a *Windows* font, so not the sort of thing I'd expect to see calling Bash.

Linux, the language of the Internet

³ I was fortunate enough to correspond with the designer, Lucas de Groot, once in relation to a legal case in which someone had used one of his fonts to forge a document. He was very helpful and kind enough to confirm the necessary facts in writing, even though he owed nothing to us.

Akamai put out a press release about the shirt when it was released, which is another sort of interesting due to the blend of tech and marketing:

Design message

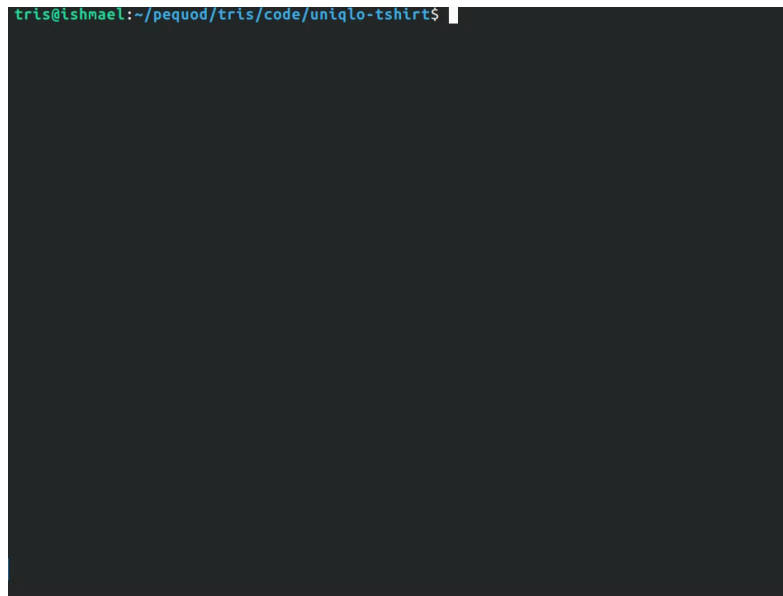
More than 25 years ago, Akamai helped make the internet we know today possible. This shirt's design is a callback to those early days of life online. The light tan color is a reference to the "beige box" plastic casings that housed the early internet computers, and the heart on the front represents how the internet has been used for good all over the world. On the back of the T-shirt is real code. It's a reference to Linux, the open-source language of the internet. This common language unites Akamai with the world's top brands and the people they serve, as we work together toward a vision of a safer and more connected world.

Not the first

I deliberately didn't search for spoilers at first, but I see that I am of course not the first person to get nerd-sniped by this. Wen Chuan Lee and that post also links to another (against which I've cross-checked my transcription above). I'm happy to carry on the chain.

Read the discussion of this post on Hacker News, which includes links to more info from

the designer and other interesting observations



© Tristan Sherliker 2025—present · tris@sherliker.net · No tracking or cookies are used on this site.