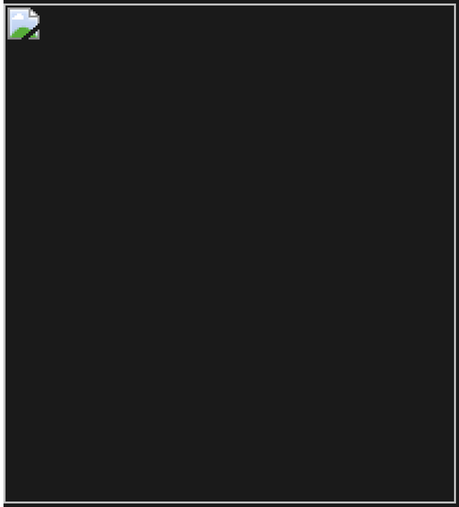
RUN YOUR DISCLOSURE WEBSITES WITH NO SERVER

by: Al Williams

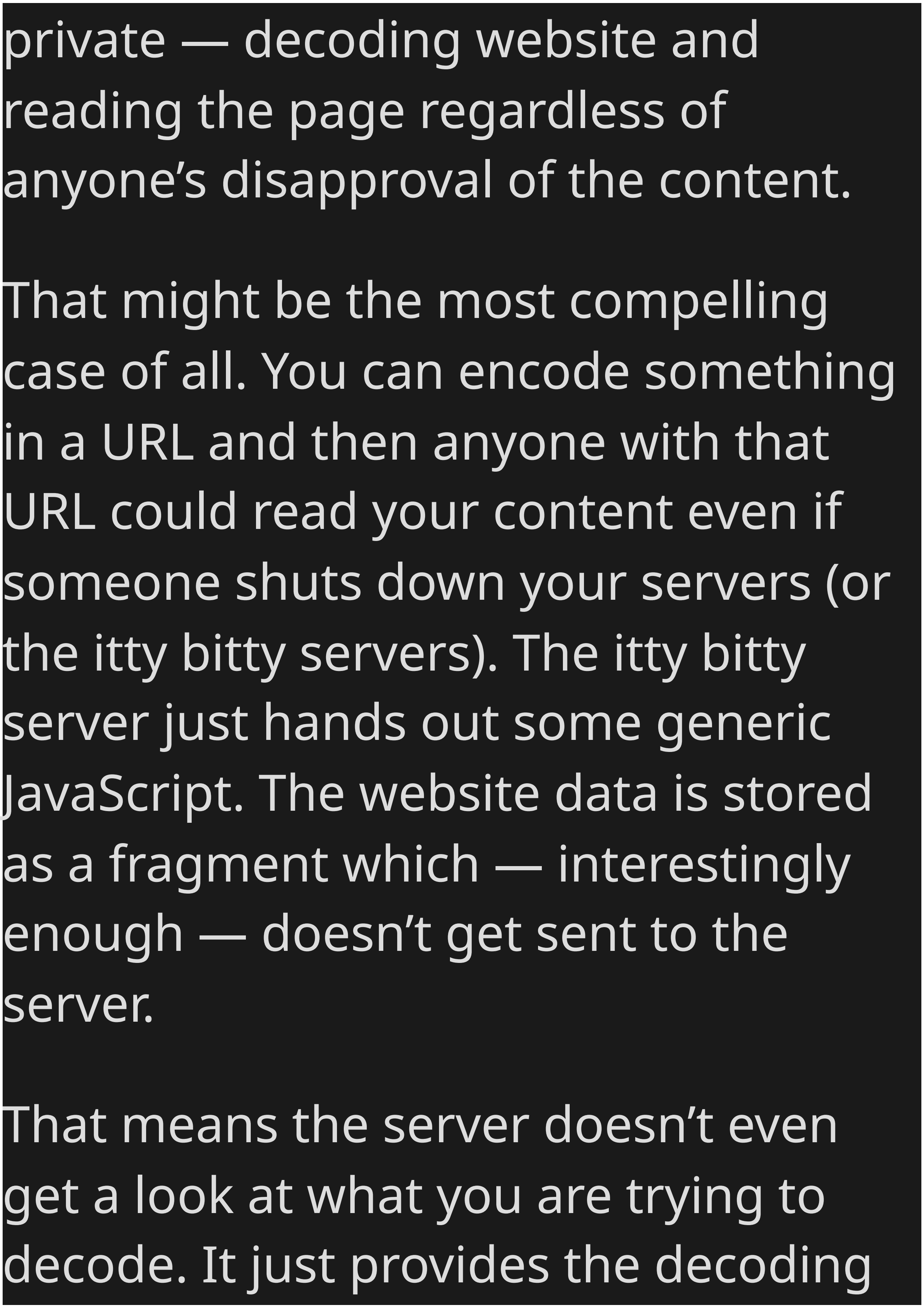
Comments

A big trend in web services right now is the so-called serverless computing, such as Amazon’s Lambda service. The idea is you don’t have a dedicated server waiting for requests for a specific purpose. Instead, you have one server (such as Amazon’s) listening for lots of requests and on demand, you spin up an environment to process that request. Conceptually, it lets you run a bit of Javascript or some other language “in the cloud” with no dedicated server. https://itty.bitty.site takes this one step farther. The site creates self-contained websites where the content is encoded in the URL itself.



Probably the best example is to simply go to the site and click on “About itty bitty.” That page is itself encoded in its own URL. If you then click on the App link, you’ll see a calculator, showing that this isn’t just for snippets of text. While this does depend on the itty.bitty.site web host to provide the decoding framework, the decoding is done totally in your browser and the code is open source. What that means is you could host it on your own server, if you wanted to.

At first, this seems like a novelty until you start thinking about it. A small computer with an Internet connection could easily formulate these URLs to create web pages. A bigger computer could even host the itty.bitty server. Then there’s the privacy issue. At first, we were thinking that a page like this would be hard to censor since there is no centralized server with the content. But you still need the decoding framework. However, that wouldn’t stop a sophisticated user from “redirecting” to another — maybe



private — decoding website and reading the page regardless of anyone's disapproval of the content.

That might be the most compelling case of all. You can encode something in a URL and then anyone with that URL could read your content even if someone shuts down your servers (or the itty bitty servers). The itty bitty server just hands out some generic JavaScript. The website data is stored as a fragment which — interestingly enough — doesn't get sent to the server.

That means the server doesn't even get a look at what you are trying to decode. It just provides the decoding framework and your browser does all the rest of the work locally. We’d love to see someone fork the project and add simple encryption, too. Currently, the text is compressed and base 64 encoded, but anyone with the URL can decode what it says. An encryption key would allow you to send URLs in the clear that only some people could decode and would be very hard to suppress.

The itty bitty code itself is an app since you can edit most pages with an edit link at the top right corner. If you don’t like editing in place, the site explains how you can use a generic HTML file or use an online HTML editor, if you prefer.

There are limitations. You probably can't host graphics internally — you'd need an external place to point to pictures. You also can make really long URLs — which means some services like Twitter will cut them off. We figure you could use a URL shortener if you needed to. There's also a way to make a QR code baked right in.

We could see this replacing a server on a Raspberry Pi project. While this isn't technically serverless computing, it did remind us of how to write code for assistants.

(Editor's Note: We've actually seen this before. Code is data, data is code.)

Posted in Network Hacks
Tagged javascript, privacy, url, web hosting, website

← Multi-Board Solder Stencils Explained

RGB Sensor's New Job: Cryptocurrency Trade Advisor →

47 THOUGHTS ON “TINY WEBSITES HAVE NO SERVER”

onebiozz says:
July 7, 2018 at 10:09 pm

oh this just reminds me of my old XSS days … i like it

Reply

Report comment

jpa says:
July 7, 2018 at 11:04 pm

Why isn’t the link just “data:text/html;base64,….”? No server necessary at all. Or even skip the base64 and you can read the content with a text editor.

One unlisted and huge limitation is that you can’t link to other “tiny sites” from one tiny webpage.

Reply

Report comment

jaap says:
July 8, 2018 at 12:01 am

When including all page data in the link, the page is actually served by the webserver that hosts the link, unless there is a way to put that link in the DNS record.

Reply

Report comment

Mishcat says:

July 8, 2018 at 12:22 am

Like this?

data:text/html;base64,SGVsbG8gSGFja2FkYXkh

Maybe it would be possible to add a tiny unpacking script to decompress a bigger page

Reply

Report comment

Greenaum says:

July 8, 2018 at 5:24 am

That worked! You can also encode graphics into URLs, a few sites do that. Must speed up loading of common images, or make them available offline or something.

This webpage-as-URL thing is a bright idea! And we don't need ANY websites to "decode" it.

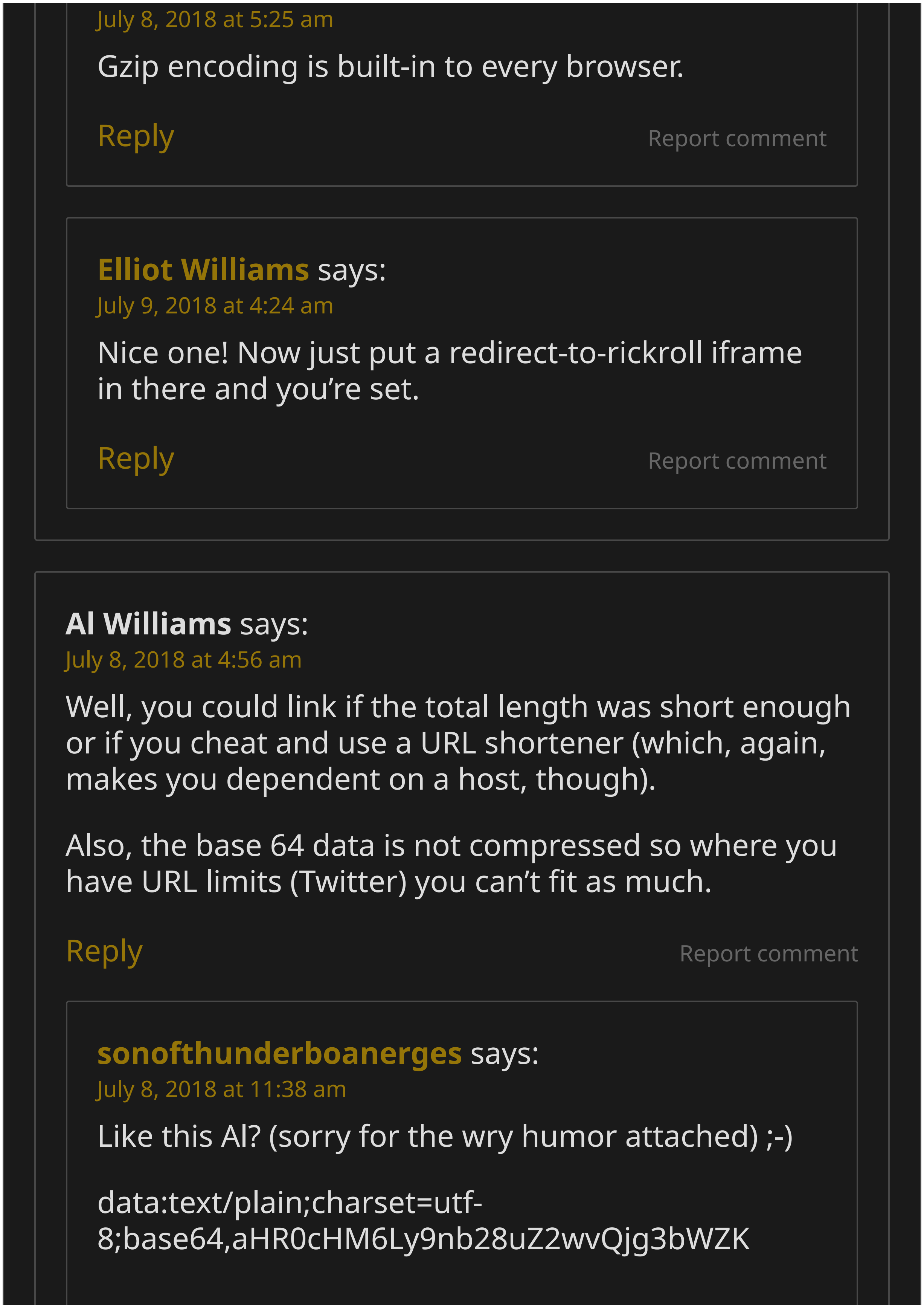
A prize for the best "site" someone posts in this thread…

[actually I haven't got a prize, but I'll applaud here in my chair]

Reply

Report comment

Ralph Doncaster says:



July 8, 2018 at 5:25 am

Gzip encoding is built-in to every browser.

Reply

Report comment

Elliot Williams says:

July 9, 2018 at 4:24 am

Nice one! Now just put a redirect-to-rickroll iframe in there and you're set.

Reply

Report comment

Al Williams says:

July 8, 2018 at 4:56 am

Well, you could link if the total length was short enough or if you cheat and use a URL shortener (which, again, makes you dependent on a host, though).

Also, the base 64 data is not compressed so where you have URL limits (Twitter) you can't fit as much.

Reply

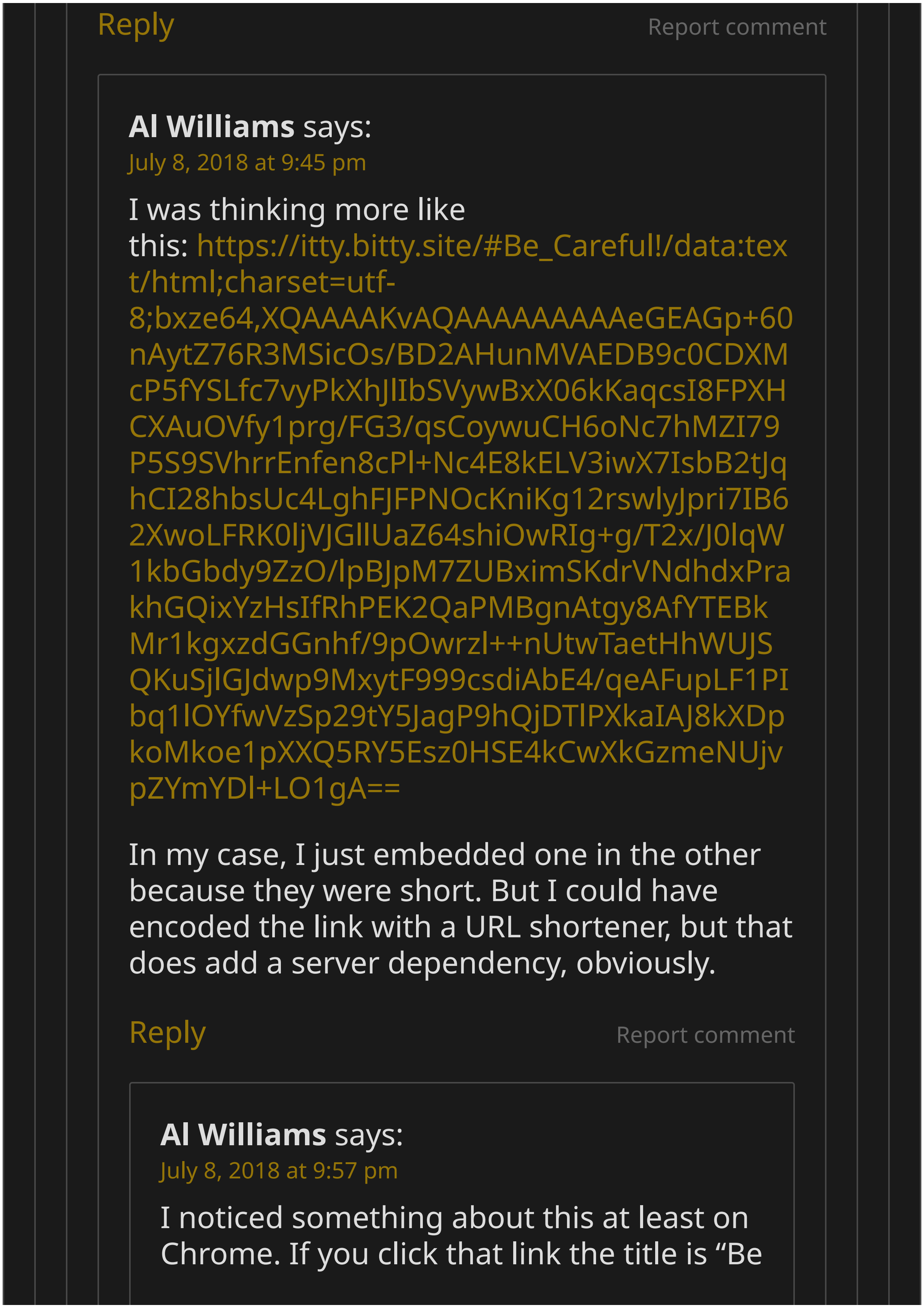
Report comment

sonofthunderboanerges says:

July 8, 2018 at 11:38 am

Like this Al? (sorry for the wry humor attached) ;-)

data:text/plain;charset=utf-8;base64,aHR0cHM6Ly9nb28uZ2wvQjg3bWZK



Al Williams says:

July 8, 2018 at 9:45 pm

I was thinking more like this: https://itty.bitty.site/#Be_Careful!/data:text/html;charset=utf-8;bxze64,XQAAAAKvAQAAAAAAAAAeGEAGp+60nAytZ76R3MSicOs/BD2AHunMVAEDB9c0CDXMcP5fYSLfc7vyPkXhJlIbSVywBxX06kKaqcsI8FPXHCXAuOVfy1prg/FG3/qsCoywuCH6oNc7hMZI79P5S9SVhrrEnfen8cPl+Nc4E8kELV3iwX7IsbB2tJqhCI28hbsUc4LghFJFPNOcKniKg12rswlyJpri7IB62XwoLFRK0ljVJGllUaZ64shiOwRIg+g/T2x/J0lqW1kbGbdy9ZzO/lpBJpM7ZUBximSKdrVNdhdxPrakhGQixYzHsIfRhPEK2QaPMBgnAtgy8AfYTEBkMr1kgxzdGGnhf/9pOwrzl++nUtwTaetHhWUJSQKuSjlGJdwp9MxytF999csdiAbE4/qeAFupLF1PIbq1lOYfwVzSp29tY5JagP9hQjDTlPXkaIAJ8kXDpkoMkoe1pXXQ5RY5Esz0HSE4kCwXkGzmeNUjvpZYmYDl+LO1gA==

In my case, I just embedded one in the other because they were short. But I could have encoded the link with a URL shortener, but that does add a server dependency, obviously.

Al Williams says:

July 8, 2018 at 9:57 pm

I noticed something about this at least on Chrome. If you click that link the title is “Be

Careful”. If you then click on the Palindrome link, the title will change to “Palindromes.” But if you hit back, you’ll get the first page but the title will not change!

Report comment

nic0mac says:

July 9, 2018 at 9:47 am

Al, hit back twice and the title will change back also…

Report comment

Valerio says:

July 8, 2018 at 11:25 pm

A possible reason could be having “data:” URLs blocked or not recognized as URLs:

https://blog.mozilla.org/security/2017/11/27/blocking-top-level-navigations-data-urls-firefox-59/

Reply

Report comment

Jac Goudsmit says:

July 7, 2018 at 11:09 pm

Oh. Oh. Here is an idea!

Print an itty bitty URL in a magazine so folks can type them into their computers at home.

Reply Report comment

bat says:
July 7, 2018 at 11:40 pm

them good old days

Reply Report comment

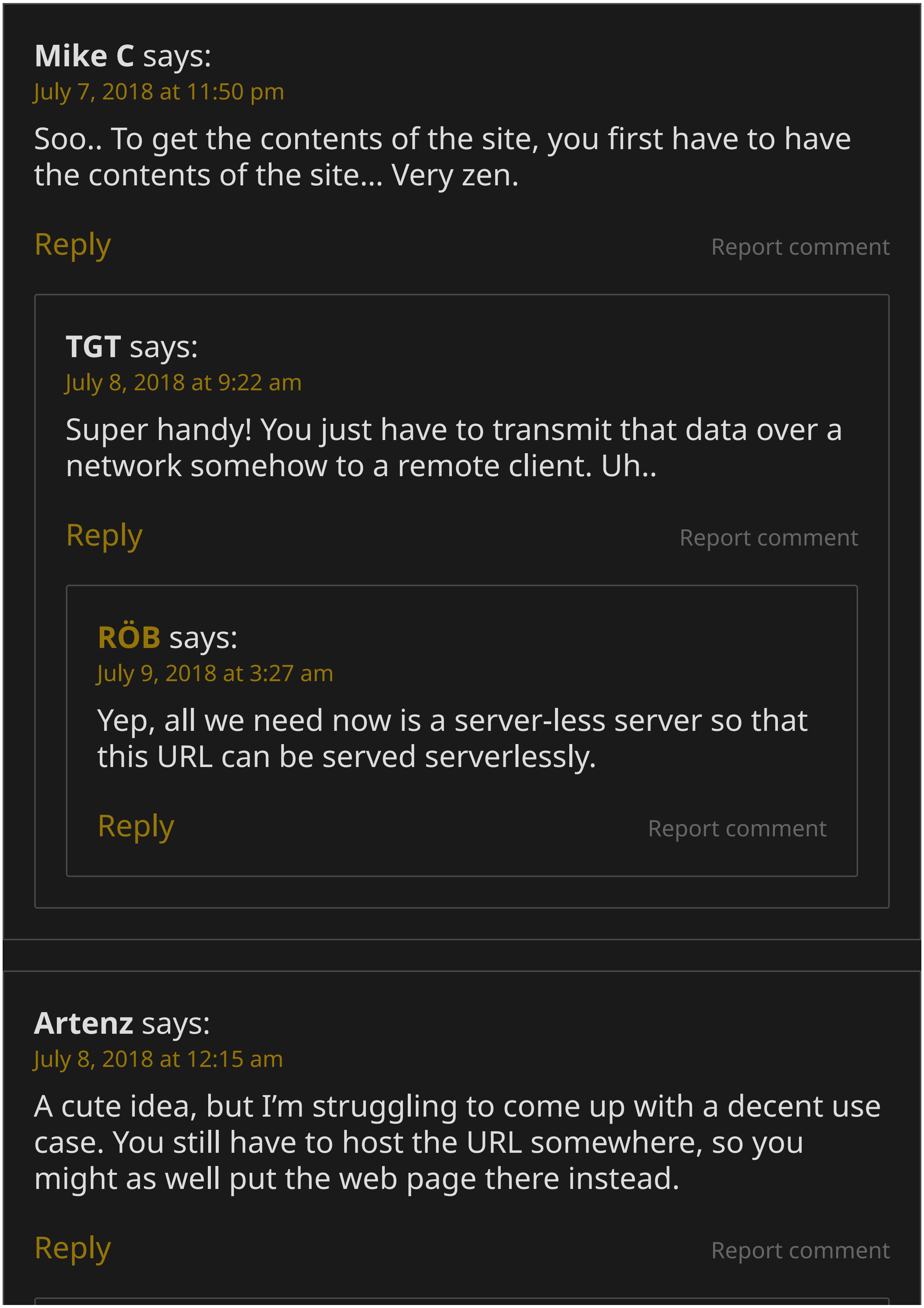
jaap says:
July 8, 2018 at 12:04 am

The good old days quickly switched to broadcasting Commodore 64 data tapes over AM radio because typing is slow. You still needed the magazine to fix up the corrupted bytes but it saved a lit of typing.

Reply Report comment

Ostracus says:
July 8, 2018 at 5:27 am

Reply Report comment



Mike C says:
July 7, 2018 at 11:50 pm

Soo.. To get the contents of the site, you first have to have the contents of the site… Very zen.

TGT says:
July 8, 2018 at 9:22 am

Super handy! You just have to transmit that data over a network somehow to a remote client. Uh..

RÖB says:
July 9, 2018 at 3:27 am

Yep, all we need now is a server-less server so that this URL can be served serverlessly.

Artenz says:
July 8, 2018 at 12:15 am

A cute idea, but I'm struggling to come up with a decent use case. You still have to host the URL somewhere, so you might as well put the web page there instead.

Nt Tn says:
July 8, 2018 at 11:59 am

I like the idea of putting these into QR codes for simple informational signs and such. It would be much more useful if the “web server” was embedded in the url too. Something like:

<a href="javascript:{document.write('testHello world.’);}”>Tiny site

Reply

Report comment

Nt Tn says:
July 8, 2018 at 12:01 pm

For the sake of argument, let’s assume hackaday’s comment system didn’t arbitrarily chop up my example which included simple html inside the document.write().

Reply

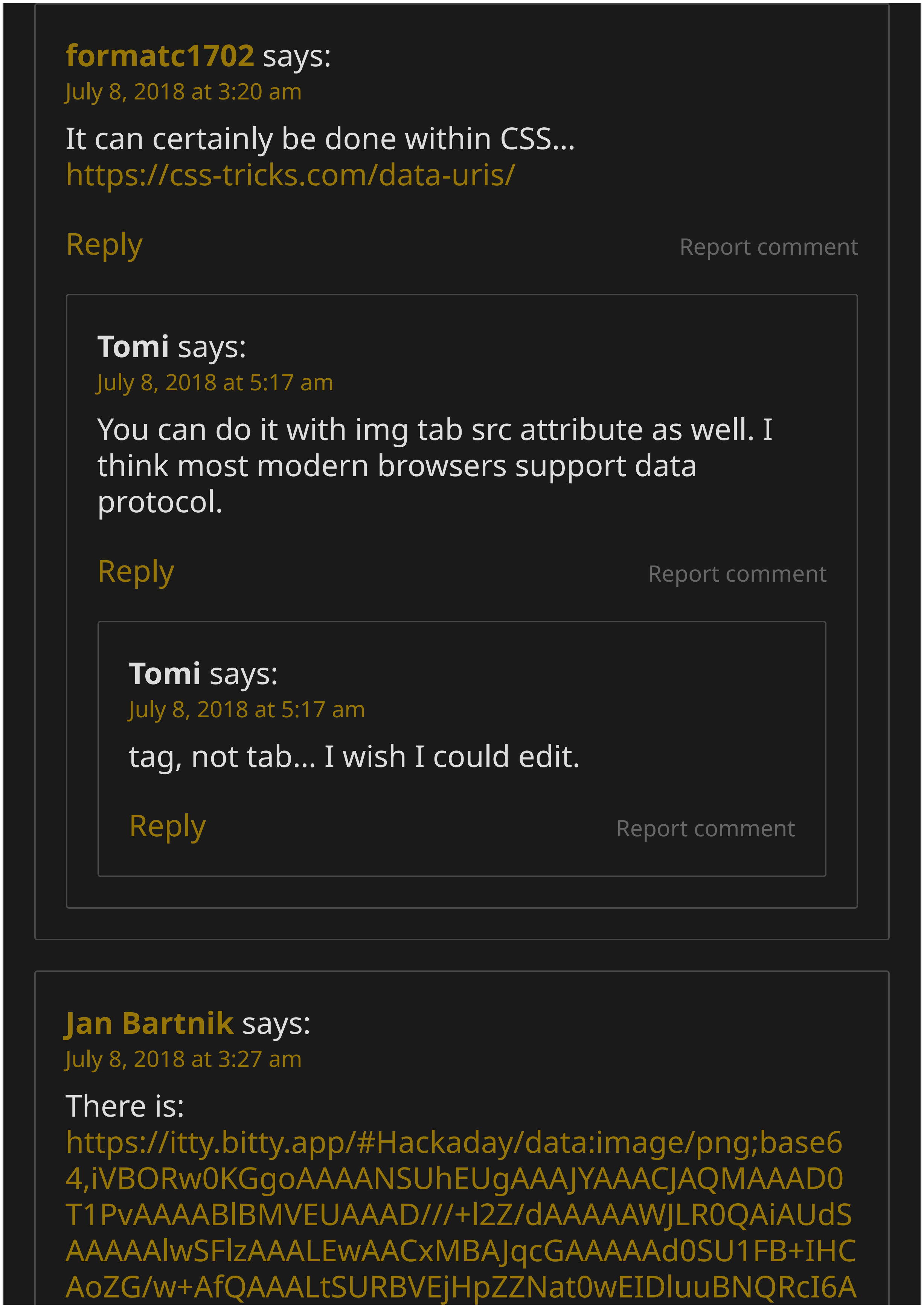
Report comment

BaGgAcFrEaK says:
July 8, 2018 at 1:20 am

Isn’t there the option to encode pictures in base64 and include in the url? I thought hak5 did that in one of their videos! Great thing!

Reply

Report comment



formatc1702 says:
July 8, 2018 at 3:20 am

It can certainly be done within CSS…
https://css-tricks.com/data-uris/

Tomi says:
July 8, 2018 at 5:17 am

You can do it with img tab src attribute as well. I think most modern browsers support data protocol.

Tomi says:
July 8, 2018 at 5:17 am

tag, not tab… I wish I could edit.

Jan Bartnik says:
July 8, 2018 at 3:27 am

There is:
https://itty.bitty.app/#Hackaday/data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAJYAAACJAQMAAAD0T1PvAAAABlBMVEUAAAD///+l2Z/dAAAAAWJLR0QAiAUdSAAAAAlwSFlzAAALEwAACxMBAJqcGAAAAAd0SU1FB+IHCAoZG/w+AfQAAALtSURBVEjHpZZNat0wEIDluuBNQRcI6A

yl24KvVWjBDl1k+Y7Qo8ThFd6yR4gfWXRZ76qF0XR+9DN
yFFLaQJ6sT/JoZqz5McbsRv/xrINFIRt4gE2xUd6a9MvgZSk
U1MWXHBTWwypiYc5siAcO6uC03sX9LCfJVmxMOkxFwd
EfH9TypJivddc2dVCMg5B0V4xtmp+xt2ZFOzSbzY3Z0B7N
FvPRePSLZqv5gv4fK7YhCWaqmO92Azhqtve7mW4BoLg
ZwoDsDJAdPeCz82b6DZAdbfH5HhkO2amOJh41AchfPU
5GWYquEsbb96wKC7I8ZlX4QGHkMVGFWBonFFrWWUa
HFohsSHI32rKM6Tw+39N0m5Je/MJKU++STXzIA/0EC9kf+
N5XljoUKx3PgFfW7CF5iyQs2ZO7aLYW7+JFGEVLCCooWD
McdnUTGeH2cuvGq7AwllvsEnPlttuzMLAlUoZLZKcSUf1d
ZHcqQk+RfdfRHZmOeBfZ+gobI9v+gU2R+f9gVfYJFAVdzS
iaNgOHLOXQiDpzUXQuJZPEDEMZyO06m9F9nY39SzZU
mXBuMjr3yEi/IyM7jowcMFTnzhQ6B7bQl1ttZRuH0HZgH
DM1W9nLrvaVXN3ap8wqn5K0H43vdmmwuxfY8Q71r7
M3LrJrYUNiTyU+LN37D/h/LhffUXy8Jx1VdSE9PpM+5RC2
gdjJFzfnuAwlA+f4hVLocpznvGEh54Mcha7kjRxxY8kv0eK
Bvw6za8prLmc5yXhripec18TXZSYre8rTKk+GlKfVPr6zEB2
afiMLPfyCR+hiQpG6ICzVBakfxL5Bqh9SWnr4SSzWGa5R
WAIucJIicI/siY9CV+MttbGWPXB5mG5D77nmWWQzq35j
PmEpRmX2wRs41tD+eV3dsK626u/WqNNro54vz+r+u2Z/
0O4tdJrKj61epdXTtJgKi3Gv+zHdp/W6vwLzYh/W6tdafV2r
/2v0iV1dDYKIqPpOkH26Pe1w3x9E/yhBSwMf1wAAAABJR
U5ErkJggg==

Reply

Report comment

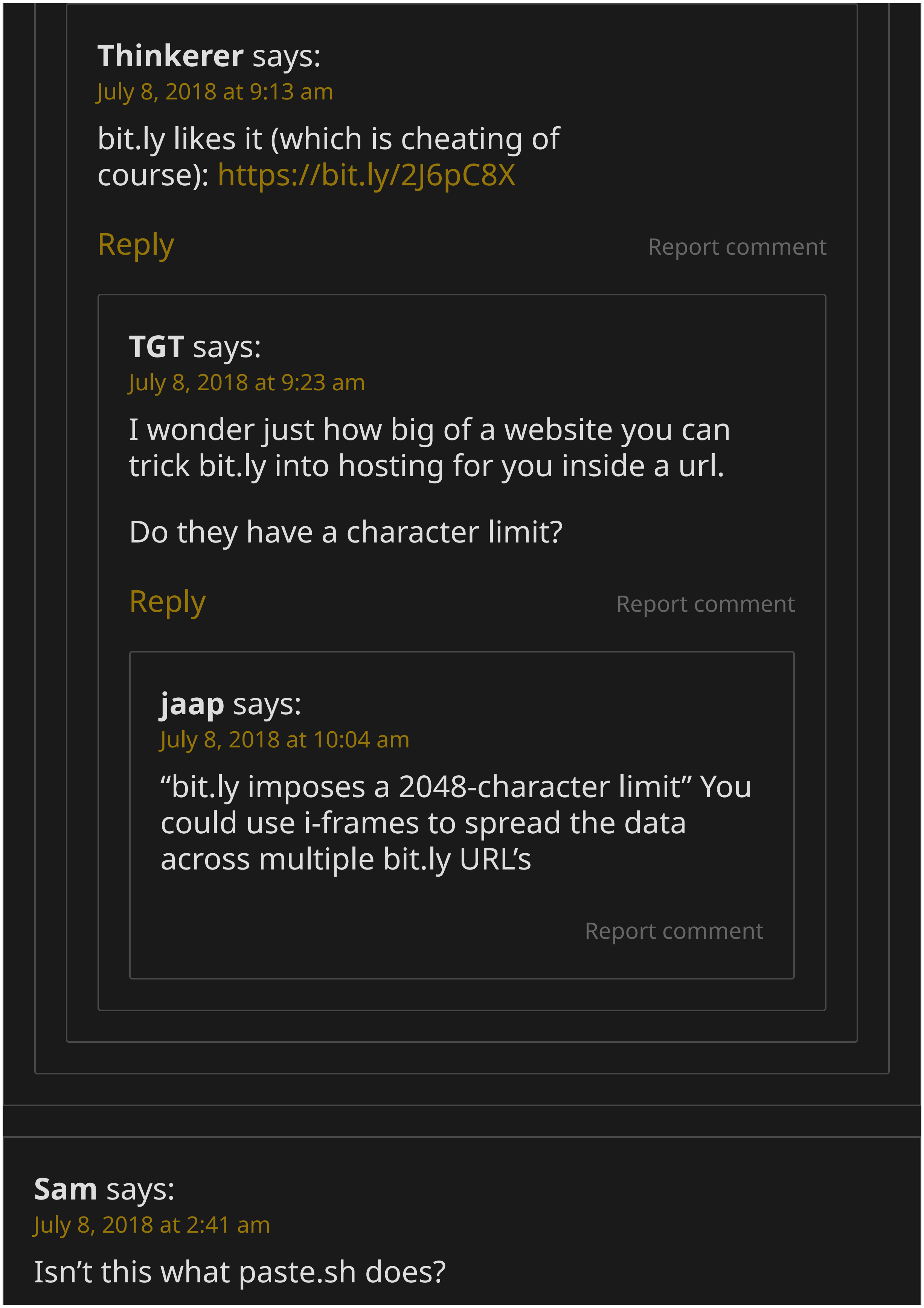
Greenaum says:

July 8, 2018 at 5:31 am

Also works fine if you cut off the server part, up to the “d” of “data”, then keep the rest. Firefox doesn’t mind data: as a URL type. That means you don’t connect to any server.

Reply

Report comment



Thinkerer says:
July 8, 2018 at 9:13 am

bit.ly likes it (which is cheating of course): https://bit.ly/2J6pC8X

TGT says:
July 8, 2018 at 9:23 am

I wonder just how big of a website you can trick bit.ly into hosting for you inside a url.

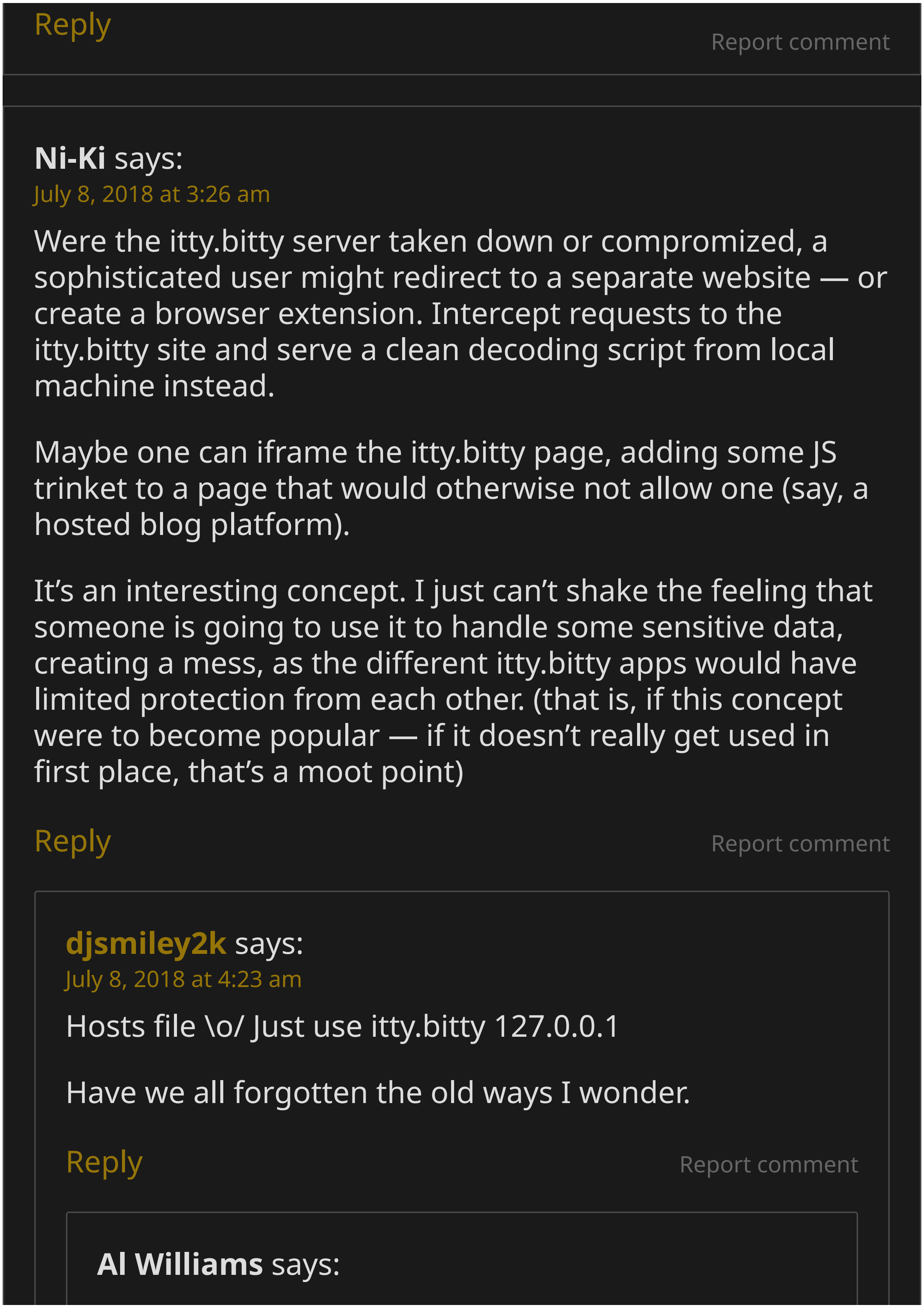
Do they have a character limit?

jaap says:
July 8, 2018 at 10:04 am

“bit.ly imposes a 2048-character limit” You could use i-frames to spread the data across multiple bit.ly URL’s

Sam says:
July 8, 2018 at 2:41 am

Isn’t this what paste.sh does?



Reply

Report comment

Ni-Ki says:

July 8, 2018 at 3:26 am

Were the itty.bitty server taken down or compromized, a sophisticated user might redirect to a separate website — or create a browser extension. Intercept requests to the itty.bitty site and serve a clean decoding script from local machine instead.

Maybe one can iframe the itty.bitty page, adding some JS trinket to a page that would otherwise not allow one (say, a hosted blog platform).

It's an interesting concept. I just can't shake the feeling that someone is going to use it to handle some sensitive data, creating a mess, as the different itty.bitty apps would have limited protection from each other. (that is, if this concept were to become popular — if it doesn't really get used in first place, that's a moot point)

Reply

Report comment

djsmiley2k says:

July 8, 2018 at 4:23 am

Hosts file \o/ Just use itty.bitty 127.0.0.1

Have we all forgotten the old ways I wonder.

Reply

Report comment

Al Williams says:

July 8, 2018 at 4:52 am

I would just replace the host name in the URL.

Reply Report comment

Ralph Doncaster (Nerd Ralph) says:
July 8, 2018 at 7:36 am

I'm not quite sure I even see the point of this. If it is to host small, basic web pages, just gzip compress the html and use Content-encoding: x-gzip, which can be done with a couple changes to the default Apache2 mime config. For example see:
http://jrdproperties.ca/old/i.html.gz
It happens to be a served from a Google compute instance, but it could just as easily be an ESP8266 in my basement. No 2K or 4K content size limit, and it only depends on one server.
I agree it's an interesting hack using the #offset in a different way than it's design intent of an index position in a web page, but I don't see how it has any practical application.

Reply Report comment

Al Williams says:
July 8, 2018 at 8:37 am

Like I said, I think the real interesting thing is if you wanted to post something you were afraid someone would try to take down. if the URL exists, it is readable. The encryption I suggested would make that even more real.

Reply

Report comment

Ralph Doncaster (Nerd Ralph) says:
July 8, 2018 at 11:23 am

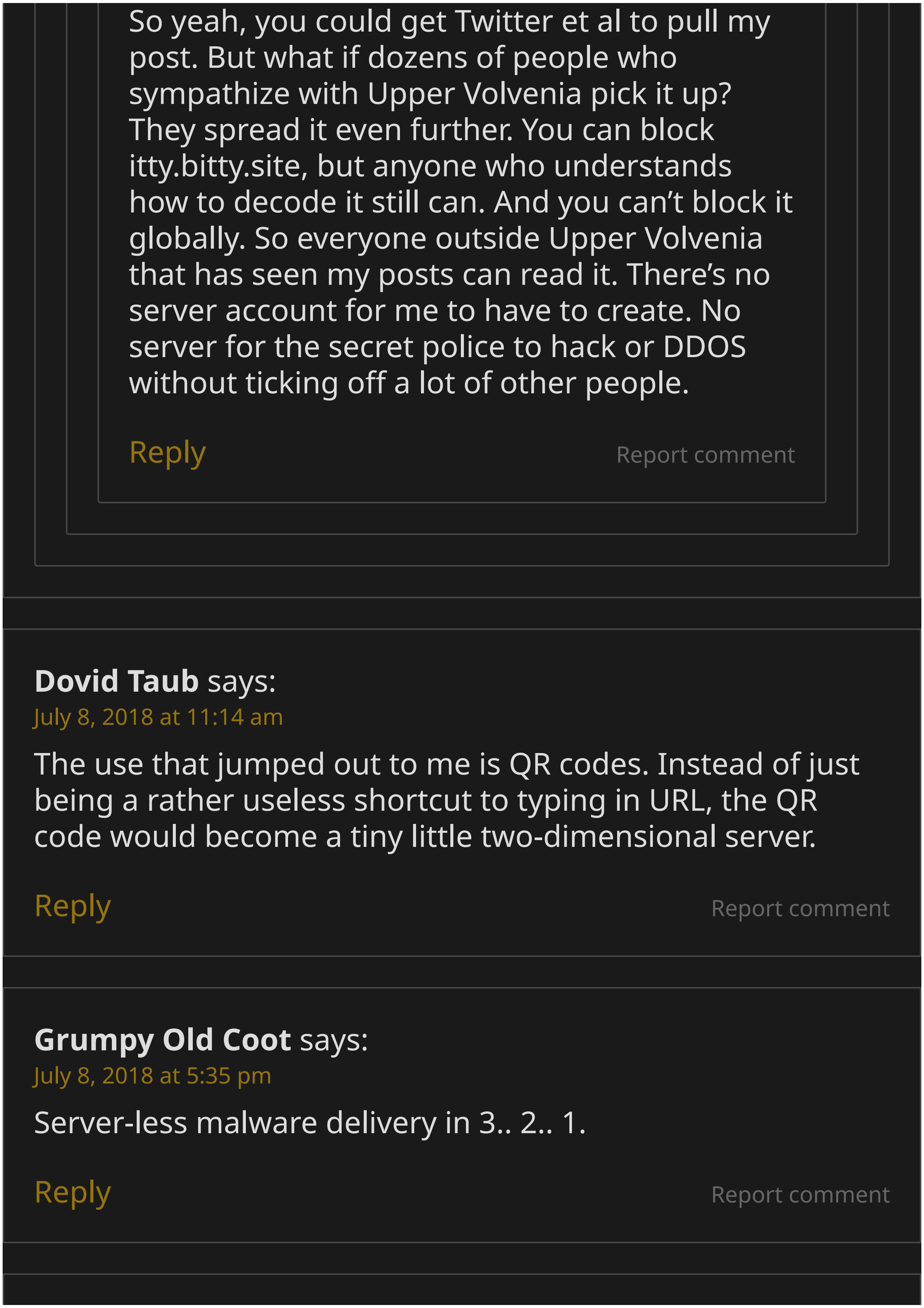
I'm no sure that's much more difficult to take down than something hosted. Twitter, google, etc often remove "objectionable" links.
The best way I know of to post something that is very difficult to purge from the internet is to post it to a newsgroup. Granted, newsgroups aren't as active as they once were, but they are still easily accessible through google groups, and there are still lots of groups with many daily posts.
https://groups.google.com/forum/#!forum/comp.lang.python

Reply

Report comment

Al Williams says:
July 8, 2018 at 9:53 pm

Yes, that's true although for many people less obvious. So I'm thinking this. I'm a political dissident in Upper Volvenia. I want to warn the world about the latest atrocities (Help! Help! I'm being repressed!). I go to an Internet cafe and I make a page up using this tool on a public PC. I post to Twitter, Reddit, Facebook, etc. I also print out the QR code 100 times. I log off and disappear after putting the QR code under all the windshields in the parking lot.



So yeah, you could get Twitter et al to pull my post. But what if dozens of people who sympathize with Upper Volvenia pick it up? They spread it even further. You can block itty.bitty.site, but anyone who understands how to decode it still can. And you can't block it globally. So everyone outside Upper Volvenia that has seen my posts can read it. There's no server account for me to have to create. No server for the secret police to hack or DDOS without ticking off a lot of other people.

Dovid Taub says:

July 8, 2018 at 11:14 am

The use that jumped out to me is QR codes. Instead of just being a rather useless shortcut to typing in URL, the QR code would become a tiny little two-dimensional server.

Grumpy Old Coot says:

July 8, 2018 at 5:35 pm

Server-less malware delivery in 3.. 2.. 1.

Ibaracat says:
July 8, 2018 at 8:16 pm

Encryption, or blockchain…

Reply

Report comment

Shay says:
July 9, 2018 at 1:03 am

Self-contained webserver? As in a text document? Want styles, then pdf. Or a .zip of a website. Or a .png or .jpg. This reminds me of the explosion of OO patterns a couple of decades ago, with lots of fancy names for things we had been doing for a long time. It's not a web server, it's just a damn web page you're sending an offline copy of to someone.

Reply

Report comment

Andras Kalmar-Nagy says:
July 9, 2018 at 5:58 am

Or you could just send the html…

Reply

Report comment

Dew-genen-ny says:
July 9, 2018 at 9:07 am

If anyone's interested, I bunged the server into a docker container to play around with:

https://hub.docker.com/r/dewgenenny/itty-bitty-docker/

Reply

Report comment

Bill says:

July 9, 2018 at 9:57 am

So a few bytes (or a few thousand) of self-contained, portable data, that can be decoded to show content on the screeen? That should have its own name. How about a “document”? (I’m still impressed, and it is very cool; time will tell how useful it is; but *new* it is not.)

Reply

Report comment