VM Workshop 2023
Securing z/VM and Linux using Tor Hidden Services

Rick Troth, CISSP, BAE Systems

<rmt@casita.net>

Audience: Linux admins, z/VM admins, z/VSE admins, cybersec aficionados, curious workshop attendees

Today’s goal: understand basic Tor concepts, see how to use Tor with z/VM, conclude “*we gotta have that!*”, use it, tell friends, stay out of trouble (IT dept, NSA)

Tor can’t help you if you don’t use it right.

disclaimer …

The content of this presentation is informational only. The reader or attendee is responsible for his/her own use of the concepts and examples presented herein.

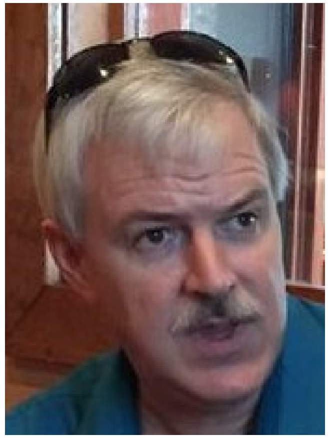
In other words: Your mileage may vary. “It Depends.”
Results not typical. Actual mileage will probably be less.
Use only as directed. Do not fold, spindle, or mutilate.
Not to be taken on an empty stomach.
Refrigerate after opening.

special disclaimer …

Many enterprises frown on this, even the presentation.
They consider Tor not suitable for corporate use.
We will show some examples.

The Workshop organizers do not want to give the idea
that they sanction using something as fringe as Tor.

about:rick



- Unix for 35+ years, Linux since 0.99
- VM/SP (et al) since 1981, VMware, Xen, KVM
- Passionate about open-source systems
- Previous jobs: SSL stack, z/VM, Unix/Linux
- Data security: Voltage 2015-2022, now at BAE

BAE SYSTEMS

BRIOSO COFFEE
BRIOSO COFFEE
French Market

Tunneling into Tor

What exactly is Tor?

- Some History, a little How-To, and stories
- Tor client proxy, tor “server”, and hidden services

What can Tor do for z/VM?

- Look outside the box (or maybe “think outside the box”)
- Leverage Tor “Hidden Services” (HS) for z/VM TCP/IP

about:tor

The Onion Router

- http://www.torproject.org/
- Originally a US Navy project, first release 2002-September-20
- Other sponsors (e.g., EFF), now 501(c)(3)

“making the web safe for whistleblowers”

about:tor

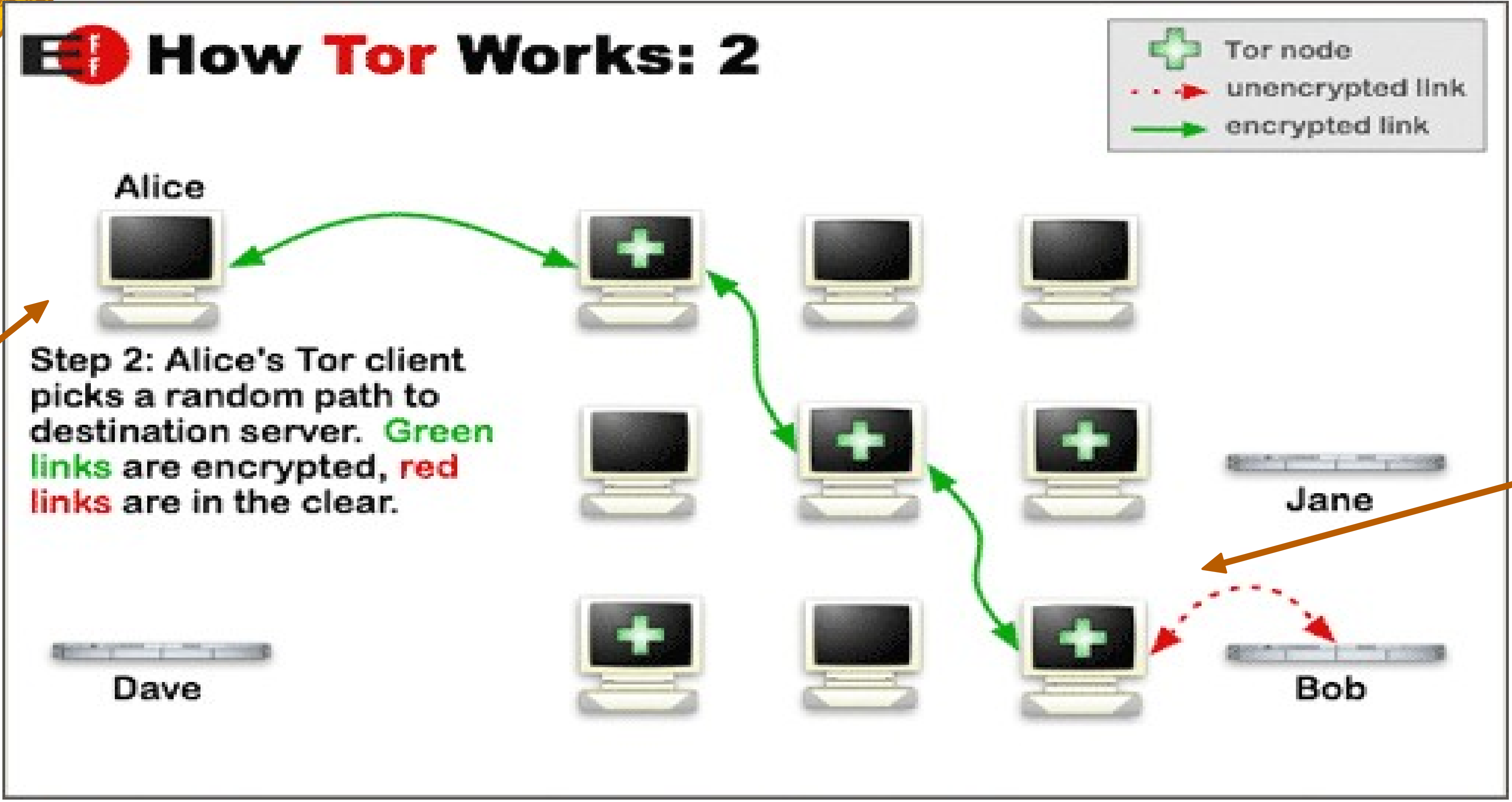
News Flash …

- July 2016 the whole Tor Project board resigned
- New board members: Matt Blaze, Cindy Cohn, Gabriella Coleman, Linus Nordberg, Megan Price, and Bruce Schneier

So what's up with that??

Can we *trust* the new board??

about:tor

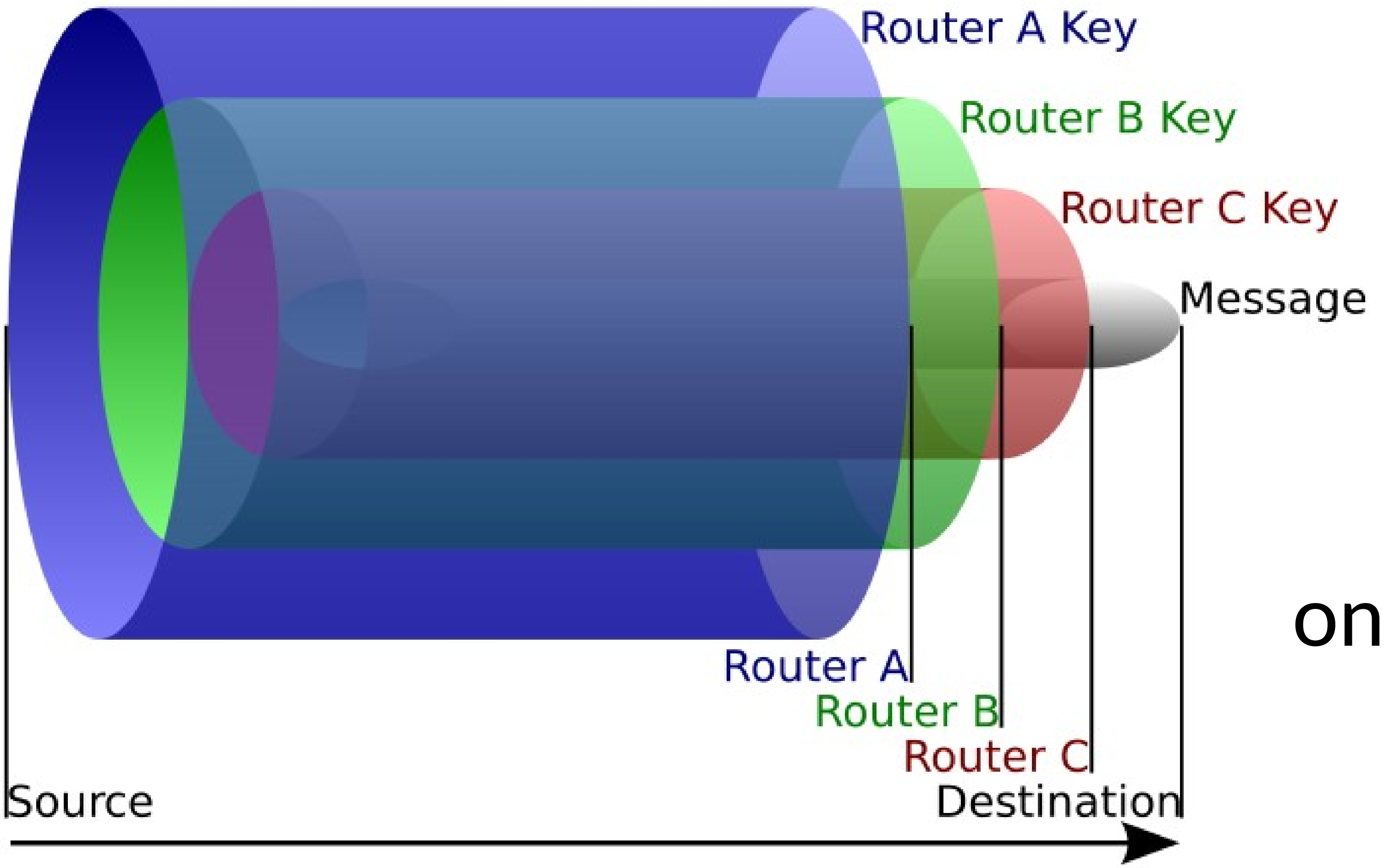


Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Your Tor

Exit node

about:tor



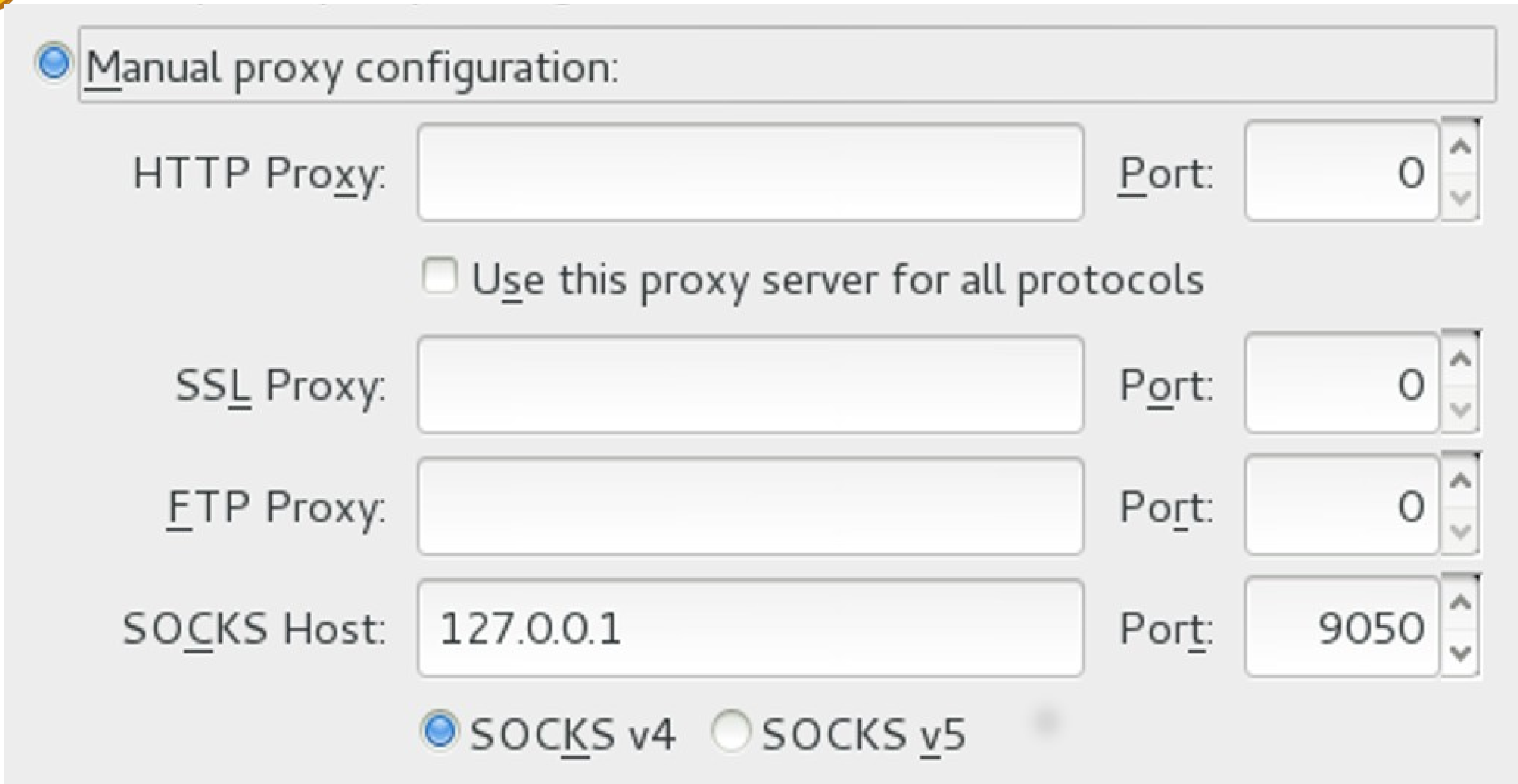
onion routing

Using Tor

“But Rick, how do we *use* it?”

- Just run it
- *Don't* run it as root
- Use an RC file, perhaps **`/etc/tor/torrc`**
  else “not present, using reasonable defaults”
- State directory **`$HOME/.tor`** will be created
- Point at it as a SOCKS proxy

Using Tor – SOCKS4a proxy



● Manual proxy configuration:

HTTP Proxy: [ ] Port: 0

□ Use this proxy server for all protocols

SSL Proxy: [ ] Port: 0

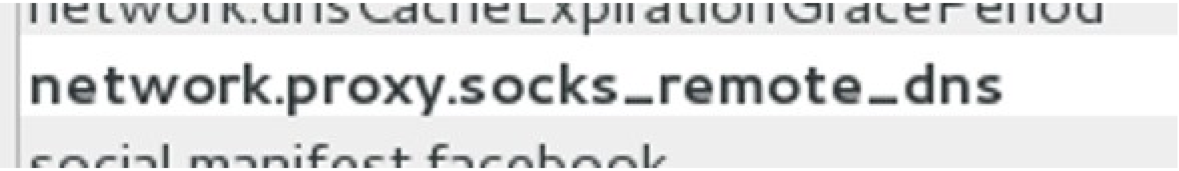
FTP Proxy: [ ] Port: 0

SOCKS Host: 127.0.0.1 Port: 9050

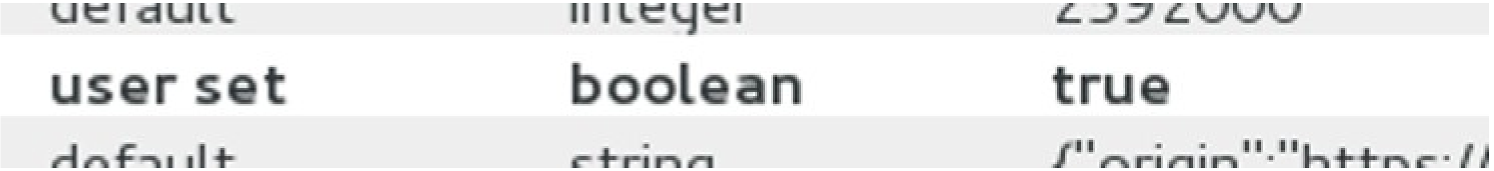
● SOCKS v4 ○ SOCKS v5

Using Tor – avoid DNS leakage

- Force hostname resolution through the proxy
- See Firefox `about:config` panel



```
network.dnsCacheExpirationGracePeriod
network.proxy.socks_remote_dns
social.manifest.facebook
```



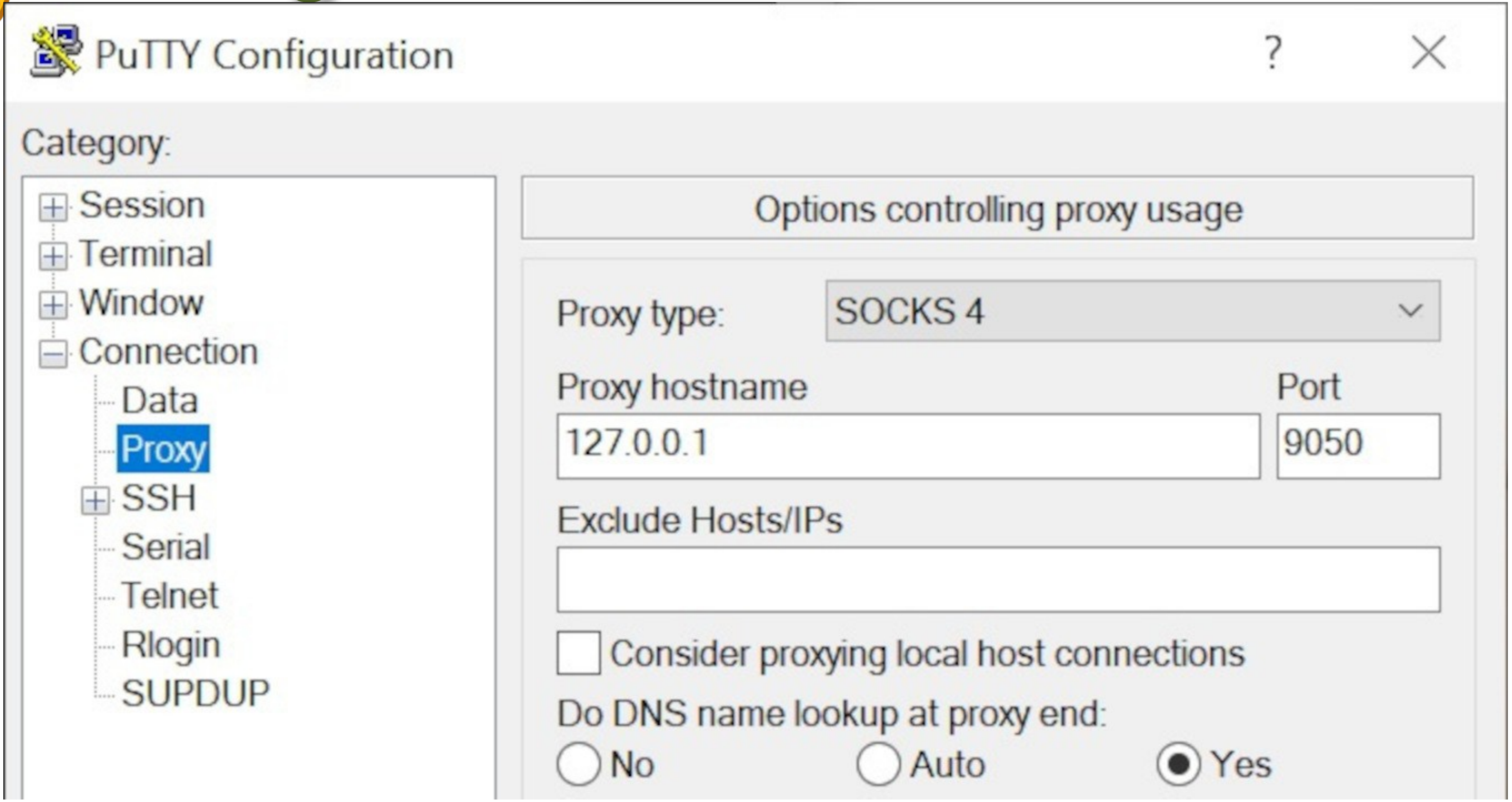
default	integer	2592000
user set	boolean	true
default	string	{"origin":"https://

Using Tor – OpenSSH and Netcat

```
ssh -o \
ProxyCommand=\
'netcat -x 127.0.0.1:9050 %h %p' \
xxxxxxxx.onion
```

Probably obvious, but it’s not all about web surfing.

Using Tor – PuTTY



PuTTY Configuration

Category:

- ⊞ Session
- ⊞ Terminal
- ⊞ Window
- ⊟ Connection
  - Data
  - Proxy
  - ⊞ SSH
  - Serial
  - Telnet
  - Rlogin
  - SUPDUP

Options controlling proxy usage

Proxy type: SOCKS 4

Proxy hostname: 127.0.0.1

Port: 9050

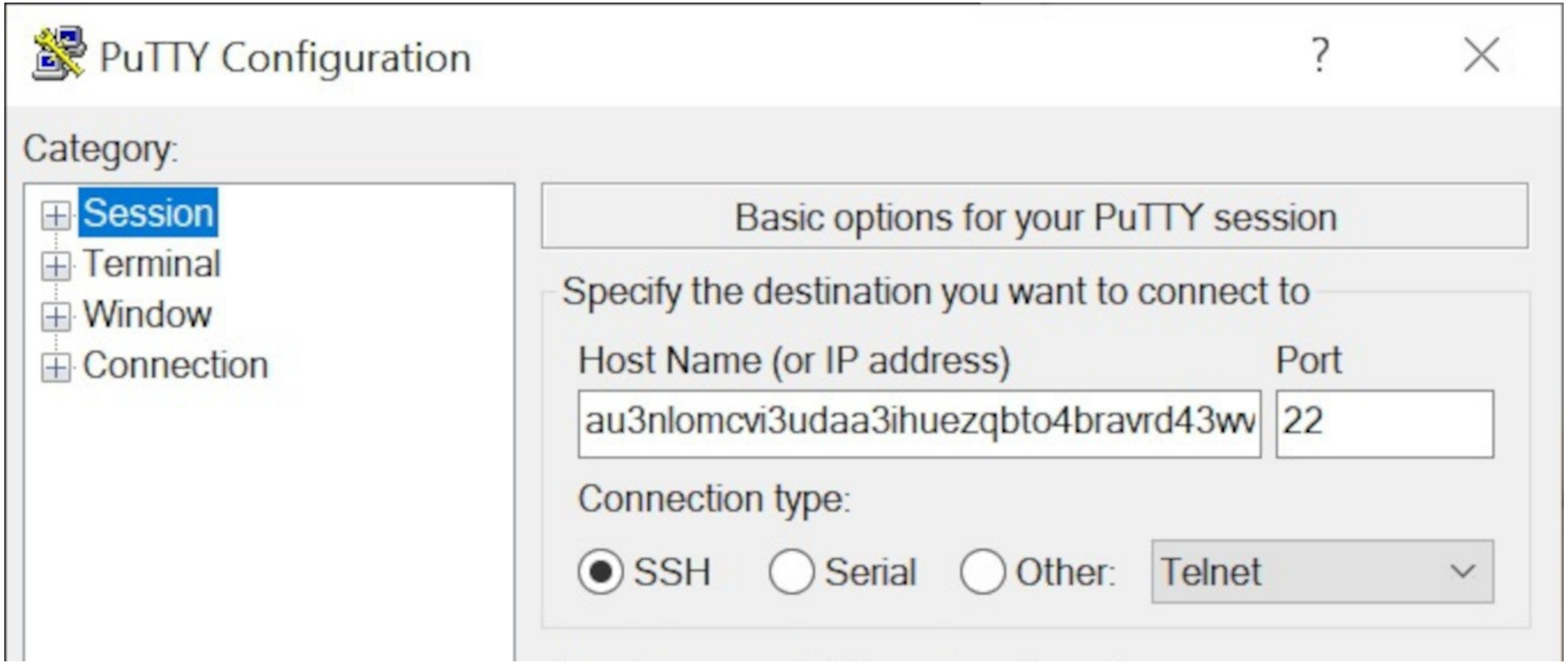
Exclude Hosts/IPs:

□ Consider proxying local host connections

Do DNS name lookup at proxy end:

○ No ○ Auto ● Yes

Using Tor – PuTTY



PuTTY Configuration

Category:

- Session
- Terminal
- Window
- Connection

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): au3nlomcvi3udaa3ihuezqbto4bravrd43wv

Port: 22

Connection type:

● SSH ○ Serial ○ Other: Telnet

Using Tor – X3270

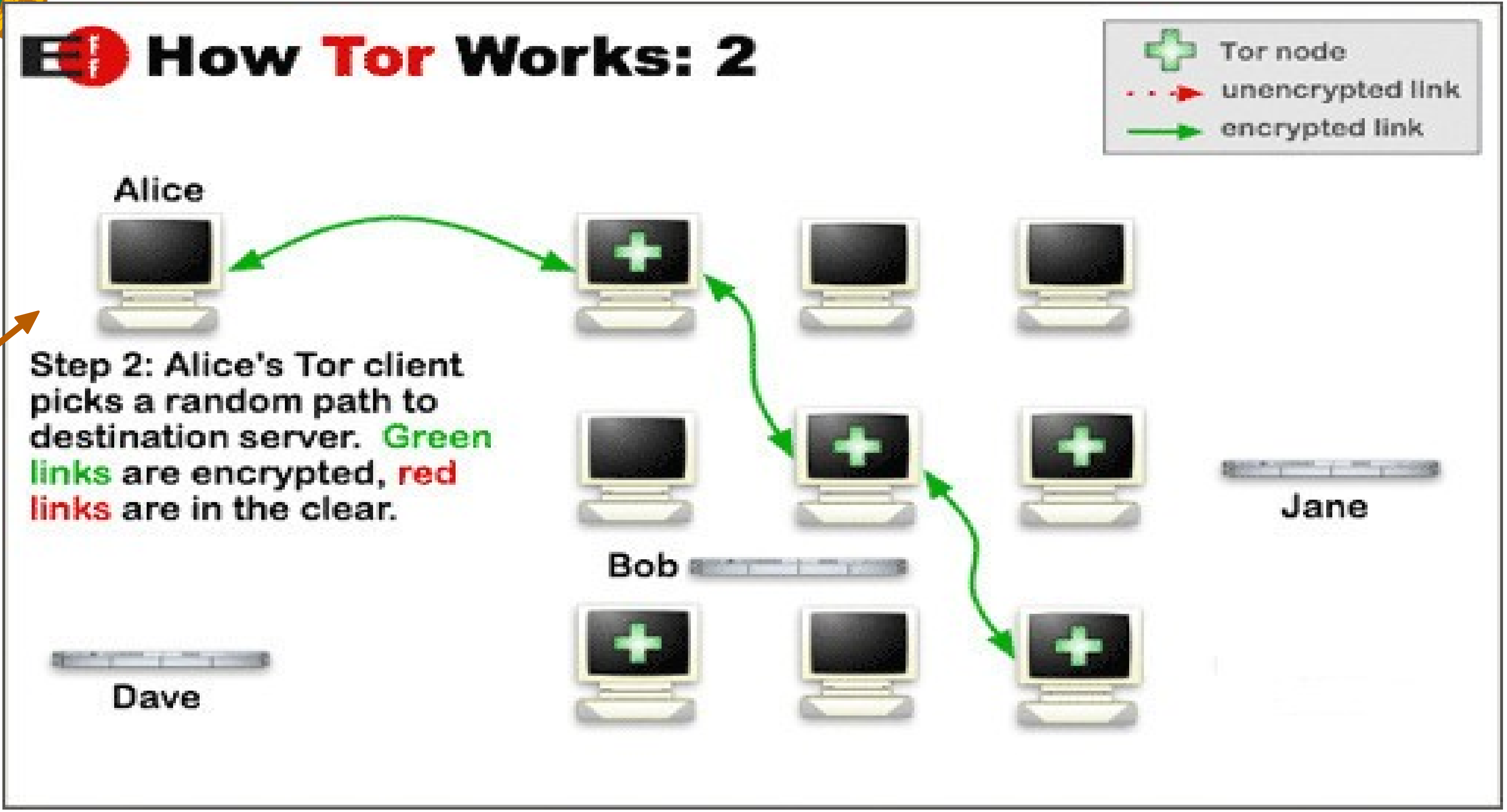
```
x3270 \
    -proxy socks4:127.0.0.1:9050 \
    xxxxxxxx.onion
```

What’s with the “dot onion”?

Introducing … *hidden services* [the crowd cheers]

- Traffic past an “exit node” is visible outside
- Traffic handled by a “hidden service” is not visible
- Hidden services are known by “`.onion`” hostnames

Where’s Bob?



Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Your Tor

No
exit node

Does It Work?

Yup, some say so.

“Not Even the NSA Can Crack

the State Dept's Favorite Anonymous Network”

[Wikipedia, Foreign Policy, “The Cable”, wayback]

Isn’t It Illegal?

Not at all, though it does get bad press.

In its filing against Ross William Ulbricht (Dread Pirate Roberts) of Silk Road, the FBI acknowledged that Tor has “known legitimate uses”.

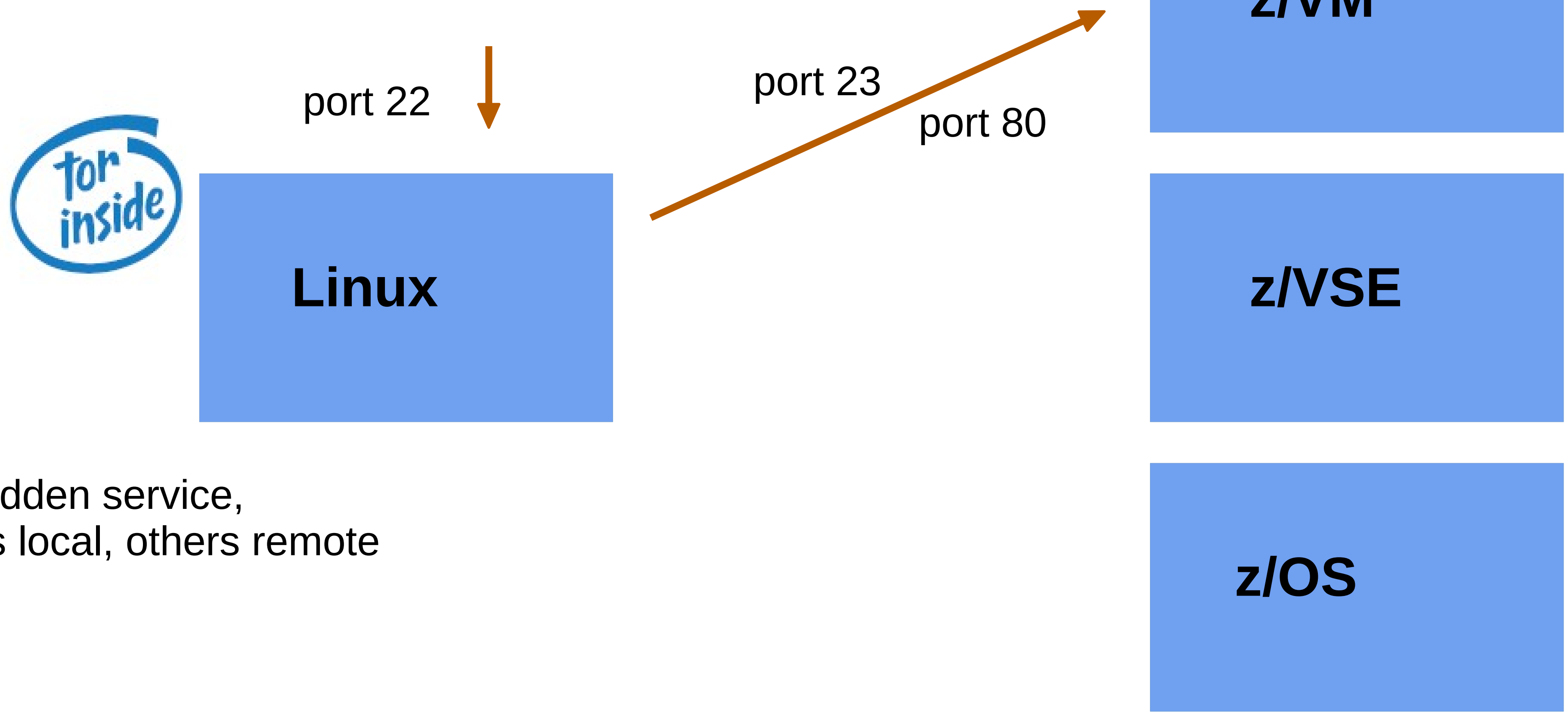
[Wikipedia, UC Berkeley, wayback]

Using Tor with z/VM

“But Rick, what’s this got to do with *VM*?”

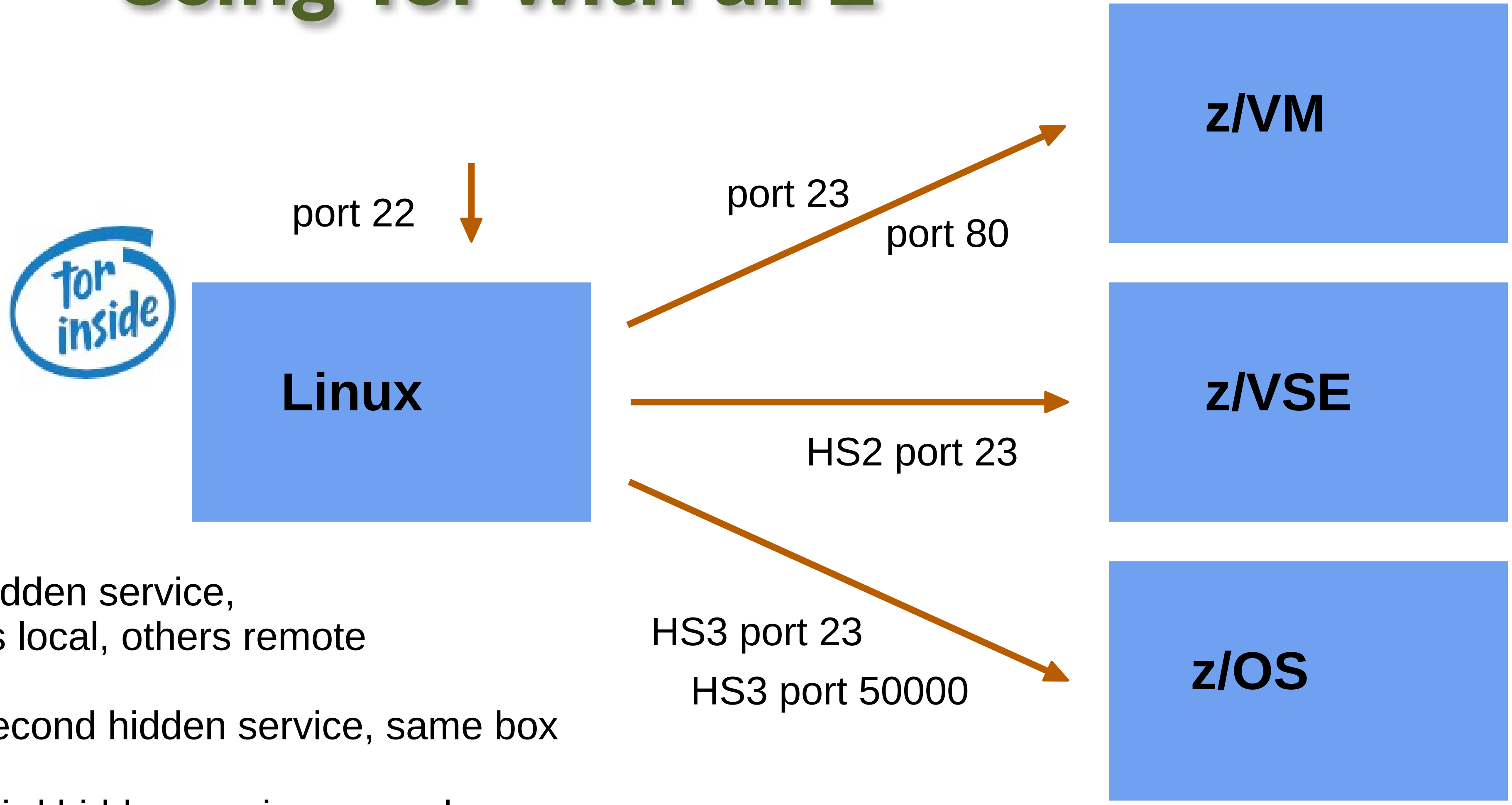
- nuthin!
- Except VM (and VSE, MVS, TPF) is in the same DC
- Use “remote” (w/r/t the Tor host) hidden services
- Use it where PKI won’t suffice; no conflict with PKI
- No changes needed to VM (nor VSE, TPF, MVS)

Using Tor with z/VM



Define a hidden service,
some ports local, others remote

Using Tor with all Z



Define a hidden service,
some ports local, others remote

Define a second hidden service, same box

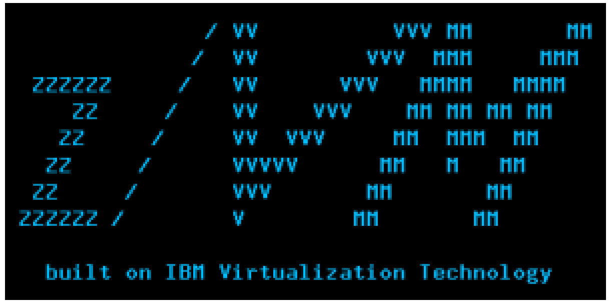
Define a third hidden service, same box

Getting Tor

- Get the source and compile it
  - `https://www.torproject.org/dist/tor-0.4.7.13.tar.gz`
  - `https://www.torproject.org/dist/tor-0.4.7.13.tar.gz.asc`
- Get it from your software package repository
  - SUSE, Debian, RedHat and derivatives, BSD

Example RC file for Tor

```
Nickname myzvmsystem
ContactInfo zVM Master <maint AT vm dot dom>
…
```

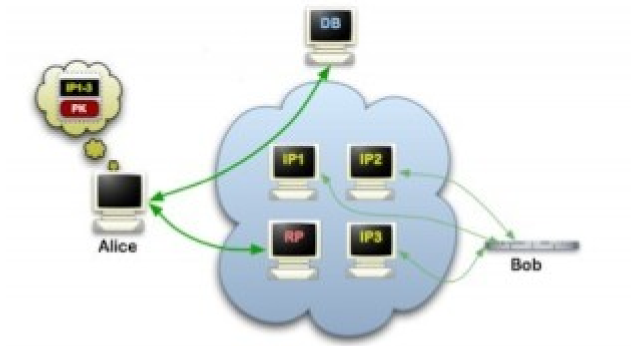


```
              / VV          VVV MM       MM
             /  VV        VVV  MMM     MMM
 ZZZZZZ     /   VV      VVV   MMMM   MMMM
    ZZ     /    VV    VVV    MM MM MM MM
   ZZ     /     VV  VVV     MM  MMM  MM
  ZZ     /      VVVVV      MM   M   MM
 ZZ     /       VVV       MM       MM
ZZZZZZ /        V        MM       MM

  built on IBM Virtualization Technology
```

Example RC file with Hidden Service

```
 …
HiddenServiceDir  /var/tor/hidden_service/
HiddenServicePort    22 192.168.29.111:22
HiddenServicePort    23 192.168.29.222:23
HiddenServicePort    80 192.168.29.222:80
```



Demo Time

.onion addresses (.onion hostnames)

The long and the short of it …

Originally: `2hiyjpes6xu5ds7l.onion`

Currently:

```
au3nlomcvi3udaa3
ihuezqbto4bravrd
43wvehyhq24ricqk
kwy2csyd.onion
```

Popular .onion Sites

- Protonmail
- Keybase
- Debian
- DuckDuckGo
- Facebook

If the site also has a public address, does it need HS?

The Pain of Certificate Management

- Generate a private key
- Generate a certificate request
- Submit the request
- … wait …
- Install the certificate, install intermediates?

- Come back next year, do it all over again

The Pain of Certificate Management

- Which CA to use?
- In-house CA needed?
- Costs of certificates justified?

There is no Easy Button

SSH, PGP, Tor, different trust models each with their own issues

The Pain of Certificate Management

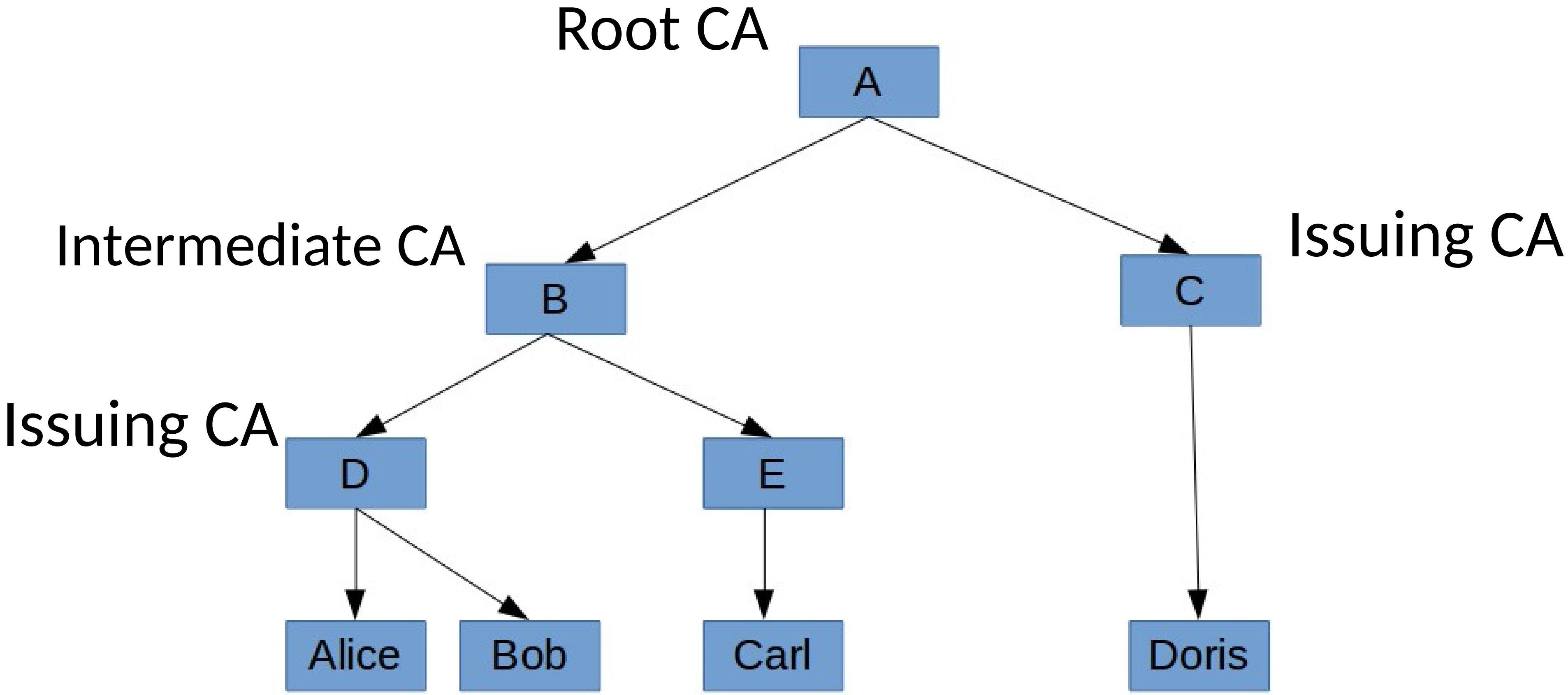
Comparing Tor with PKI

- No CA to trust (but must trust the Tor network)
- No certificate to manage (hidden service key is automatic)
- Full anonymization (connections are not easily tracked)

The Pain of Certificate Management

One of the inherent problems of standard HTTPS is that trust put in a website is defined by certificate authorities: a hierarchical and closed set of companies and governmental institutions approved by your web browser vendor. This model of trust has long been criticized and proven … to be vulnerable to attacks …

Trust Models



Root CA

A

Intermediate CA

B

Issuing CA

C

Issuing CA

D

E

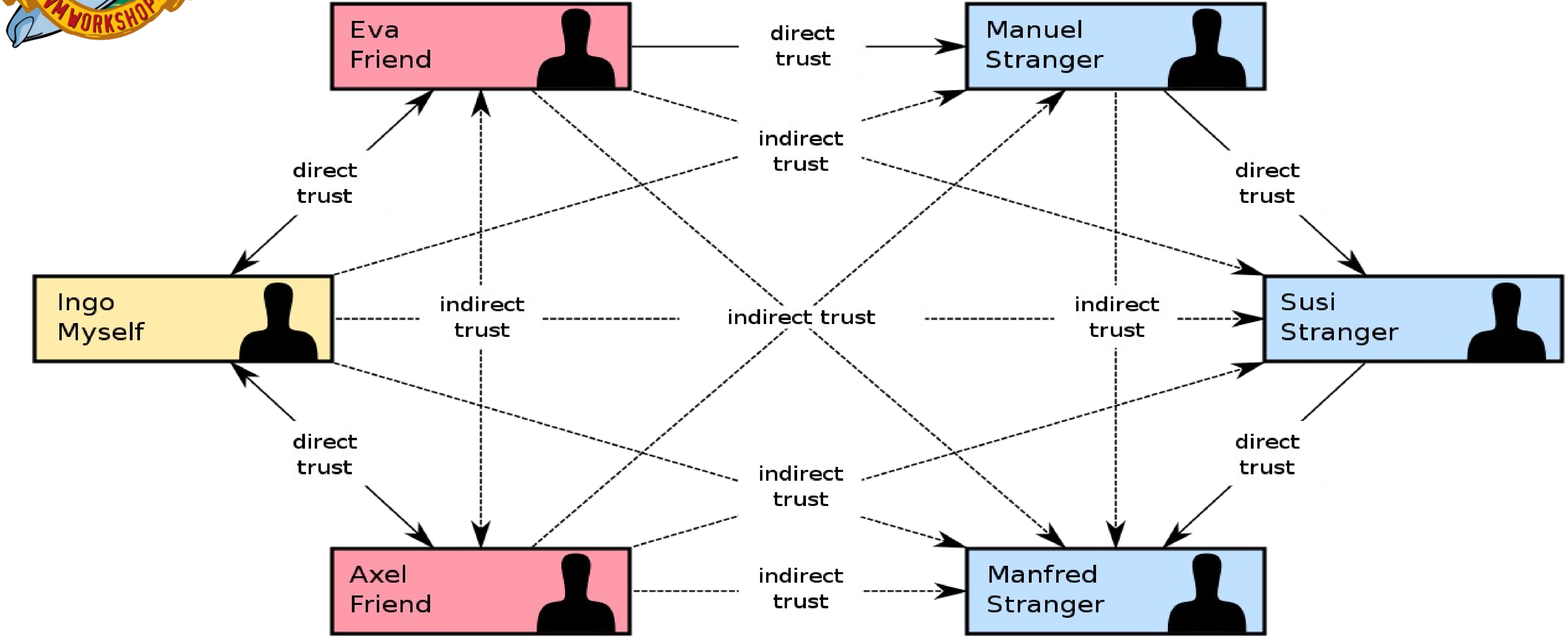
Alice

Bob

Carl

Doris

Trust Models



Exit Node Extras

Three kinds of Tor nodes

- Exit Node (or “exit relay”, seen above)
- Relay Node (“guard” or “middle”, generally safe from worry)
- Bridge Node (unpublished exit)

This was
my mistake

Hidden services
live here

Exit Node Extras

To run an exit node, in your “`torrc`” file:

- `ORPort 9001`
- `DirPort 9030`

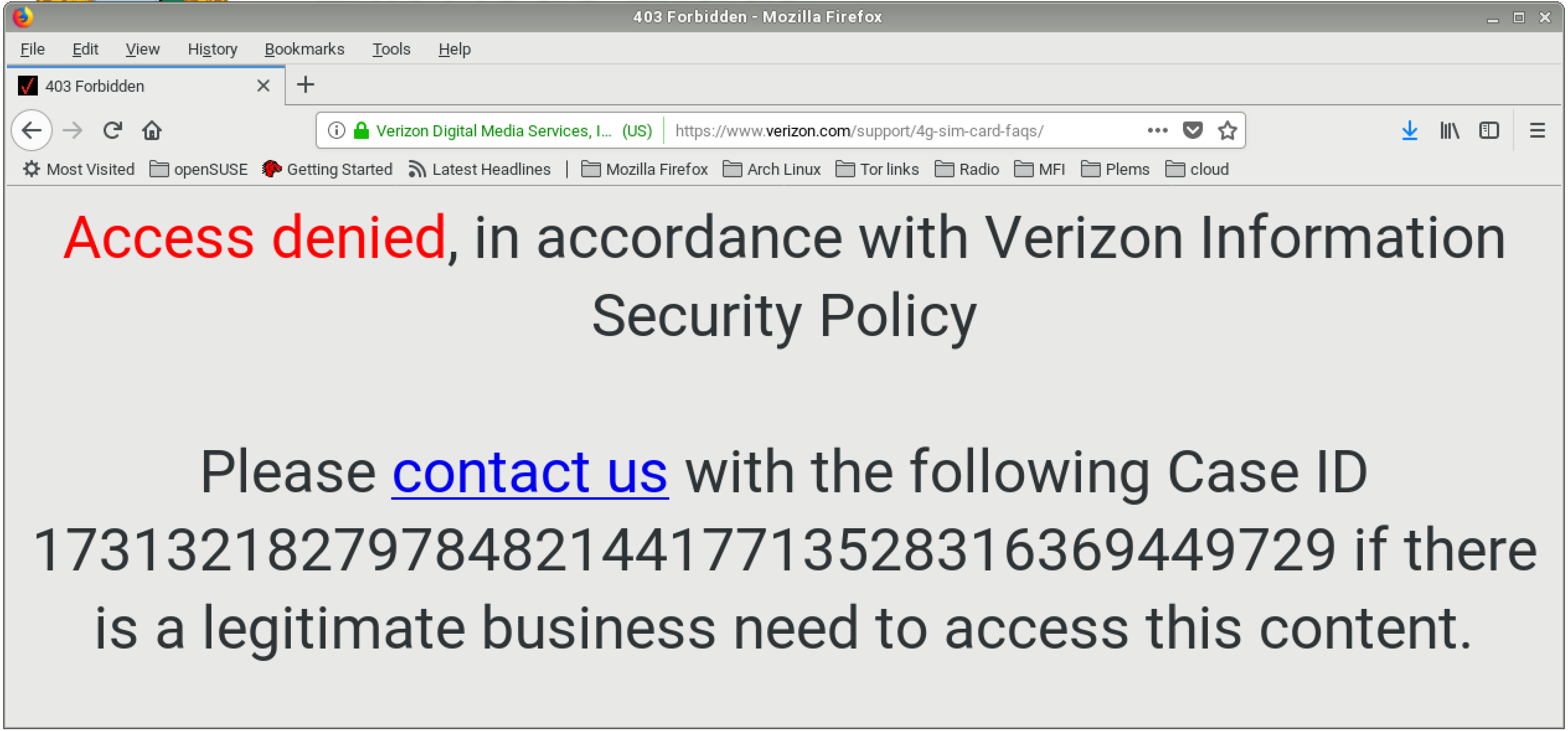
2020, doing research for a Tor talk,
I left an exit node running … at home

Exit Node Extras

Home residential IP address suddenly blocked by …

- Key Bank,
- Capital One credit card issuer,
- Verizon Wireless cellular provider,
- Norwegian Cruise Line,
- Zoom conferencing

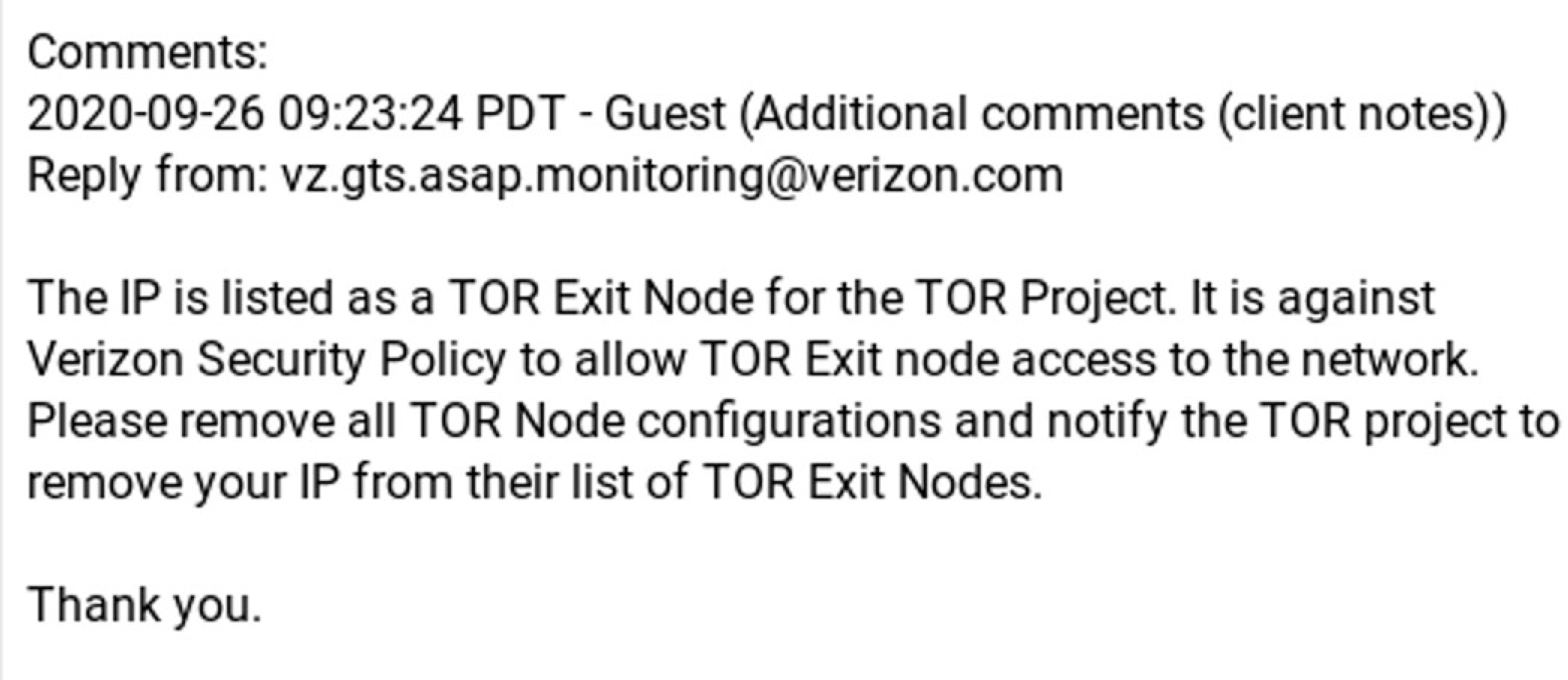
Exit Node Extras



Access denied, in accordance with Verizon Information Security Policy

Please contact us with the following Case ID 173132182797848214417713528316369449729 if there is a legitimate business need to access this content.

Exit Node Extras



Comments:
2020-09-26 09:23:24 PDT - Guest (Additional comments (client notes))
Reply from: vz.gts.asap.monitoring@verizon.com

The IP is listed as a TOR Exit Node for the TOR Project. It is against Verizon Security Policy to allow TOR Exit node access to the network. Please remove all TOR Node configurations and notify the TOR project to remove your IP from their list of TOR Exit Nodes.

Thank you.

Exit Node Extras

Further response …

“Although some users of the TOR project are using it for good intentions and so forth, it is also a place where nefarious users can also perform anonymous malicious attacks and attempt fraudulent activities. Thus, Verizon deems the project's network as risky and restricts communications from their TOR Exit Nodes. As a security specialist, we hope you can understand this position. We apologize for the inconvenience. However, it is for the security of all our Verizon clients.”

Conclusion … maybe *you* should use Tor

- Tor is a tool providing anonymity (privacy)
- Tor Hidden Services provide strong end-to-end encryption
  - do not interfere with other security protocols (e.g., TLS)
  - do not require changes to VM or sibling systems
- Tor is easy to run and configure and relatively easy to use

Thank you!

`http://www.casita.net/vmworkshop/2023/torforzvm.ppt`

`http://www.casita.net/vmworkshop/2023/torforzvm/`

Thank you!

Or when you’re “on” Tor …

`[]=au3nlomcvi3udaa3ihuezqbto4bravrd43wvehyhq24ricqkkwy2csyd.onion`

`http://[]/vmworkshop/2023/torforzvm.ppt`

`http://[]/vmworkshop/2023/torforzvm/`

Building Tor from Source

“If you’re not using the source code, then ‘open source’ might not really be part of your supply chain.”

package-*version*`.tar.`*zz*

package-*version*`.tar.`*zz*`.asc` (or `.sig`, `.sign`)

`https://www.torproject.org/dist/tor-0.4.7.13.tar.gz`

Getting and Vetting the Source

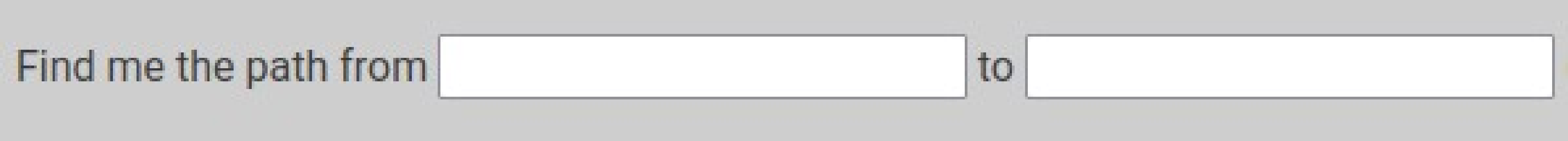
```
gpg --verify package-version.tar.zz.asc
```

- Extract the key ID (check the sig, it will fail)
- Find that key in the Web-of-Trust
- Walk the trust chain; if trusted then add key
- Check the signature again (for real)

Getting and Vetting the Source

- Get files, extract key, find in WOT, follow the chain
- Do you trust it? If so then add key and re-check src sig
- Signing key: `0x42e86a2a11f48d36`

`https://the.earth.li/~noodles/pathfind.html`

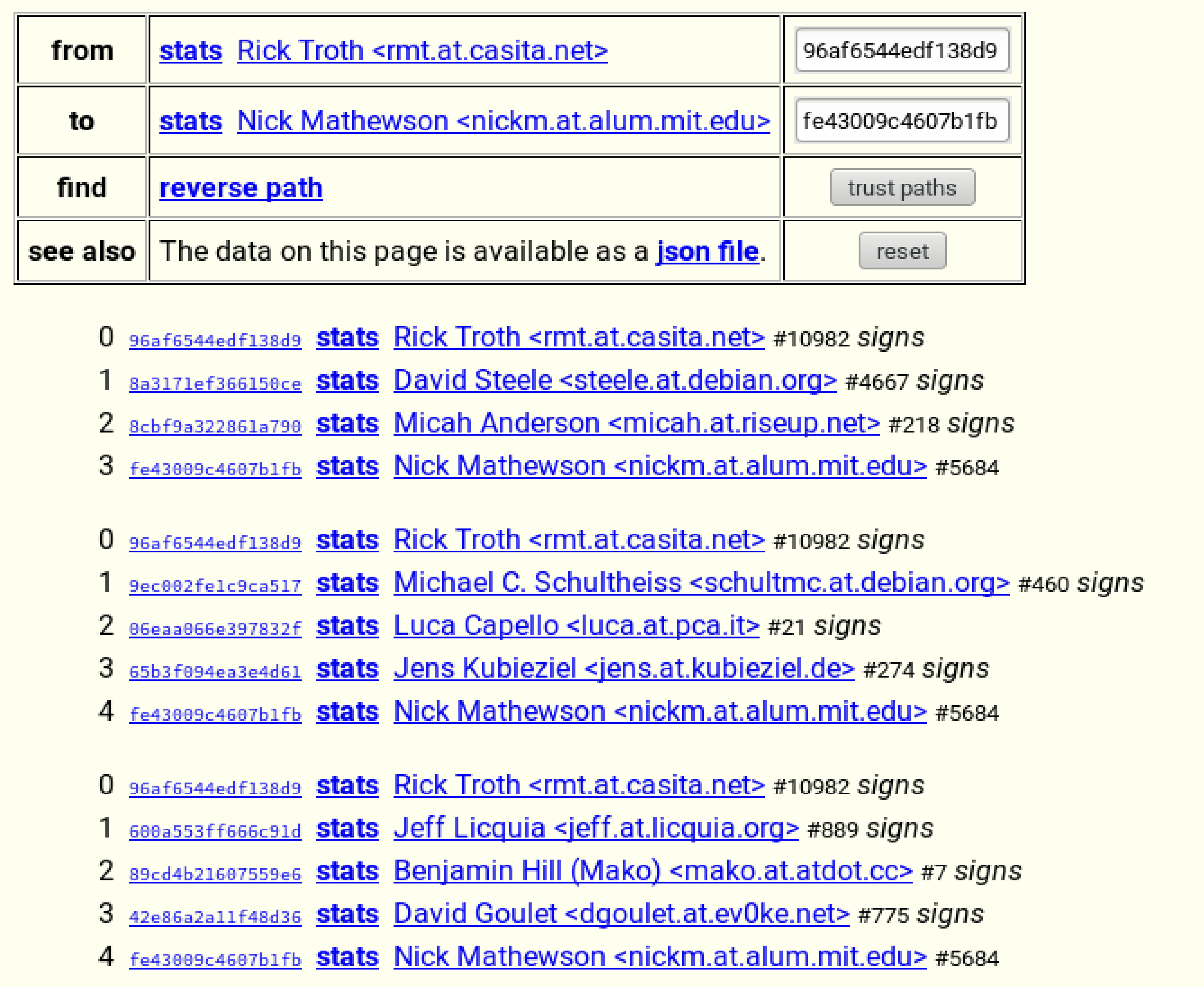


Find me the path from to

Tor project sometimes signs a hash and not the tarball.

Getting and Vetting the Source

Multiple “paths” between the keys provide more assurance.



from	**stats** Rick Troth <rmt.at.casita.net>	96af6544edf138d9
to	**stats** Nick Mathewson <nickm.at.alum.mit.edu>	fe43009c4607b1fb
find	**reverse path**	trust paths
see also	The data on this page is available as a **json file**.	reset

0 96af6544edf138d9 **stats** Rick Troth <rmt.at.casita.net> #10982 *signs*
1 8a3171ef366150ce **stats** David Steele <steele.at.debian.org> #4667 *signs*
2 8cbf9a322861a790 **stats** Micah Anderson <micah.at.riseup.net> #218 *signs*
3 fe43009c4607b1fb **stats** Nick Mathewson <nickm.at.alum.mit.edu> #5684

0 96af6544edf138d9 **stats** Rick Troth <rmt.at.casita.net> #10982 *signs*
1 9ec002fe1c9ca517 **stats** Michael C. Schultheiss <schultmc.at.debian.org> #460 *signs*
2 06eaa066e397832f **stats** Luca Capello <luca.at.pca.it> #21 *signs*
3 65b3f094ea3e4d61 **stats** Jens Kubieziel <jens.at.kubieziel.de> #274 *signs*
4 fe43009c4607b1fb **stats** Nick Mathewson <nickm.at.alum.mit.edu> #5684

0 96af6544edf138d9 **stats** Rick Troth <rmt.at.casita.net> #10982 *signs*
1 600a553ff666c91d **stats** Jeff Licquia <jeff.at.licquia.org> #889 *signs*
2 89cd4b21607559e6 **stats** Benjamin Hill (Mako) <mako.at.atdot.cc> #7 *signs*
3 42e86a2a11f48d36 **stats** David Goulet <dgoulet.at.ev0ke.net> #775 *signs*
4 fe43009c4607b1fb **stats** Nick Mathewson <nickm.at.alum.mit.edu> #5684

Explode, Config, “just make”

```
tar xzf tor-0.4.7.13.tar.gz (then ‘cd’)
./configure  optional --prefix=
make
make install
make clean   or make distclean
```

(or use Chicory)