Securing z/VM and Linux using Tor Hidden Services

Rick Troth

2023 June 23, VM Workshop, Columbus

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!*"

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.

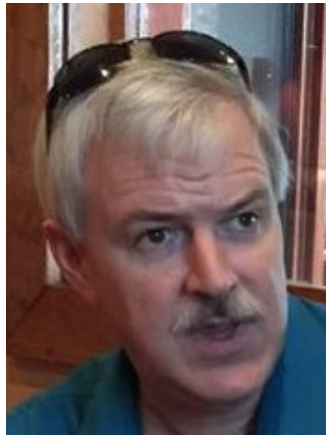
In other words: Your mileage may vary. “It Depends.” Results not typical. Actual mileage will probably be less. Use only as directed. Do not fold, spindle, or mutilate. Not to be taken on an empty stomach. Refrigerate after opening.

special disclaimer …

Many enterprises frown on things which are considered fringe. They consider Tor not suitable for corporate use. We will show some examples of “Tor hate” and how to respond to it.

Tor is a powerful, yet easy-to-use tool.

about:rick



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

"It's all about trust!"

BAE SYSTEMS

Voltage security

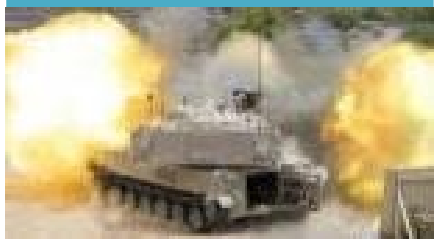
MICRO FOCUS

BAE Systems, Inc. – a commanding breadth of capabilities

At BAE Systems, we serve, supply, and **protect those who protect us** in a company culture that is performance-driven and values-led. We aim to provide a vital advantage to our customers through our world-class defense capabilities across the air, land, maritime, and cyber domains, and by accelerating the pace of innovation. As one of the broadest and most geographically diverse international defense companies, we have an important role to play in contributing to the economic prosperity of the places where our people live and work, and in using our knowledge and technologies to reduce the environmental impacts of our operations.

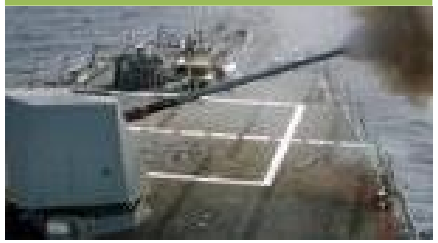
Electronics

Land

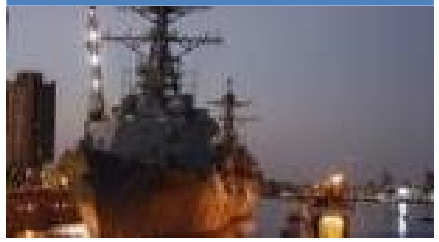


Intelligence

Maritime



Support Services



Tunneling into Tor

What exactly is Tor?

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

What can Tor do for z/VM?

- Leverage Tor “Hidden Services” (HS) for z/VM TCP/IP
- Reachability without Risk

about:security

- Privacy != Security and vice versa
- Anonymity != Privacy, but they do overlap
- Good algorithms are at the mercy of bad implementations
- Good implementations are at the mercy of bad deployment
- It’s all about trust

about:security

Security vs Usability, Closeted vs Networked + Hardened

- Don’t react in fear; Consider the impact of the choices
- Compare with redundant systems, components, services

We don’t have (and *should* have) redundant secure methods

“Security is a Process”

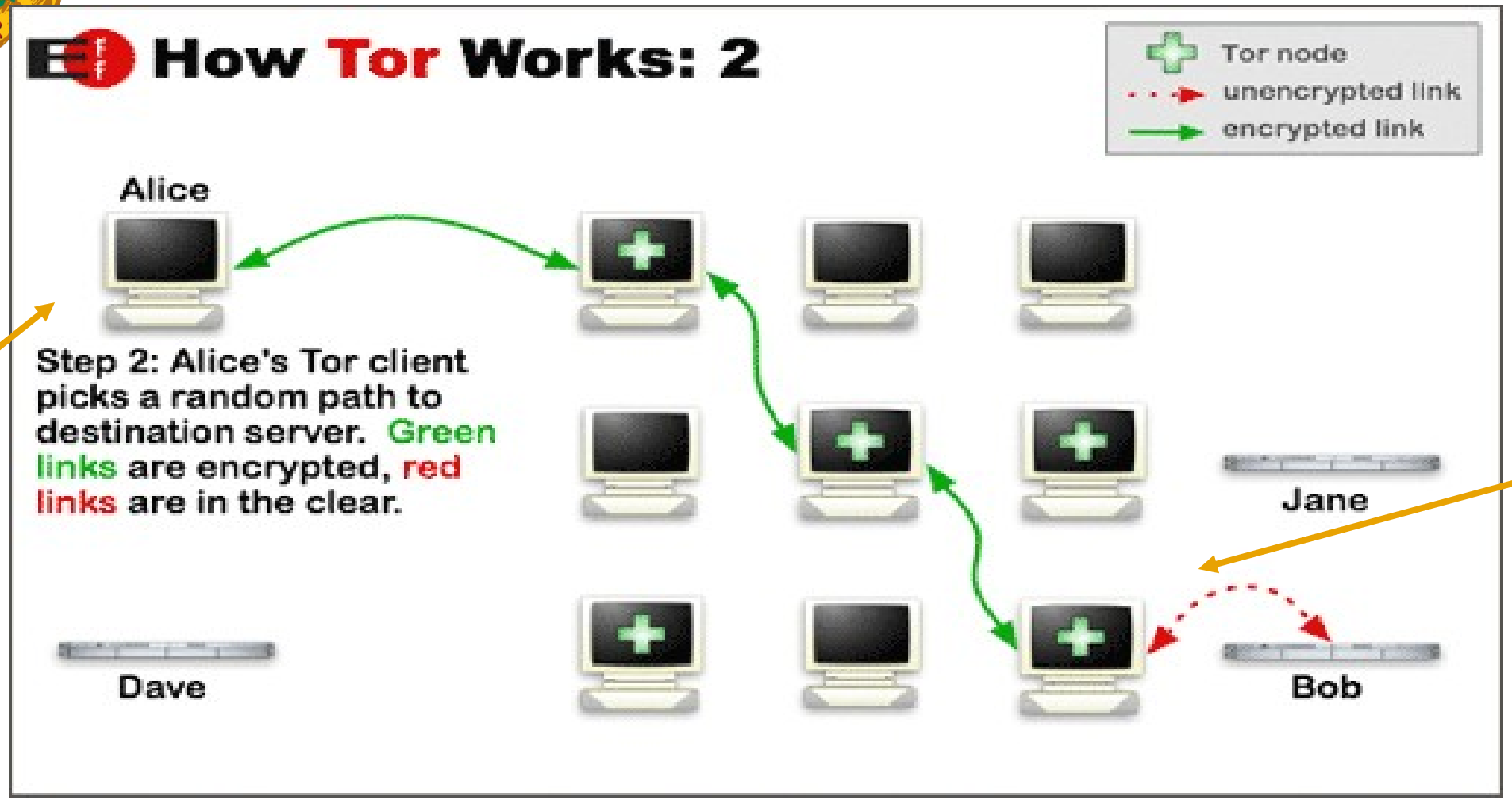
about:tor

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"

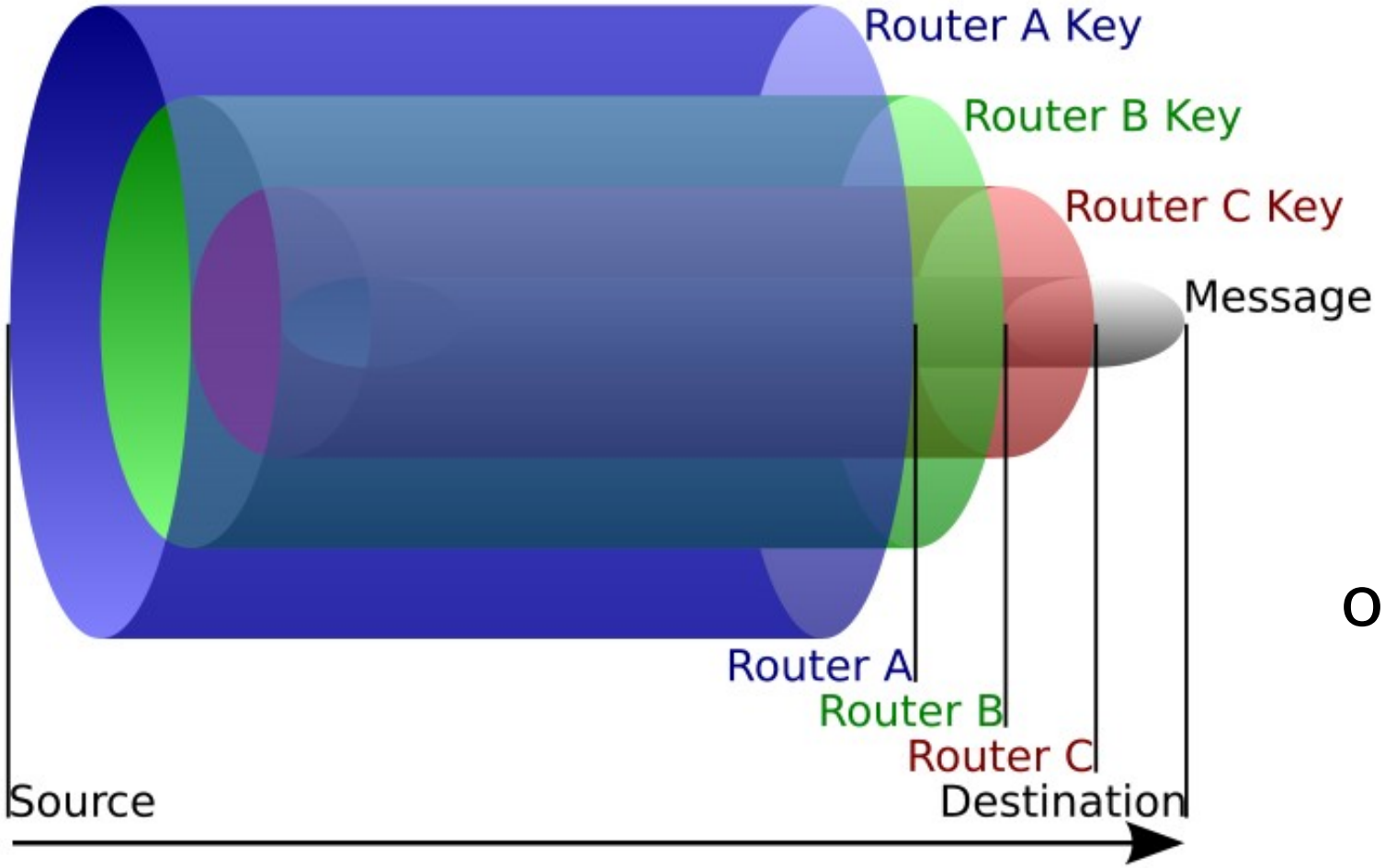
about:tor



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, localhost port 9050

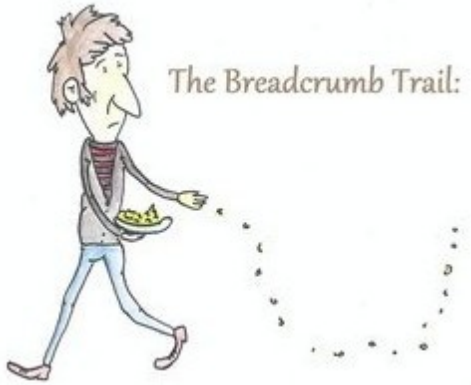
Using Tor – SOCKS4a proxy

Using Tor – avoid DNS leakage

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

network.proxy.socks_remote_dns

user set	boolean	true

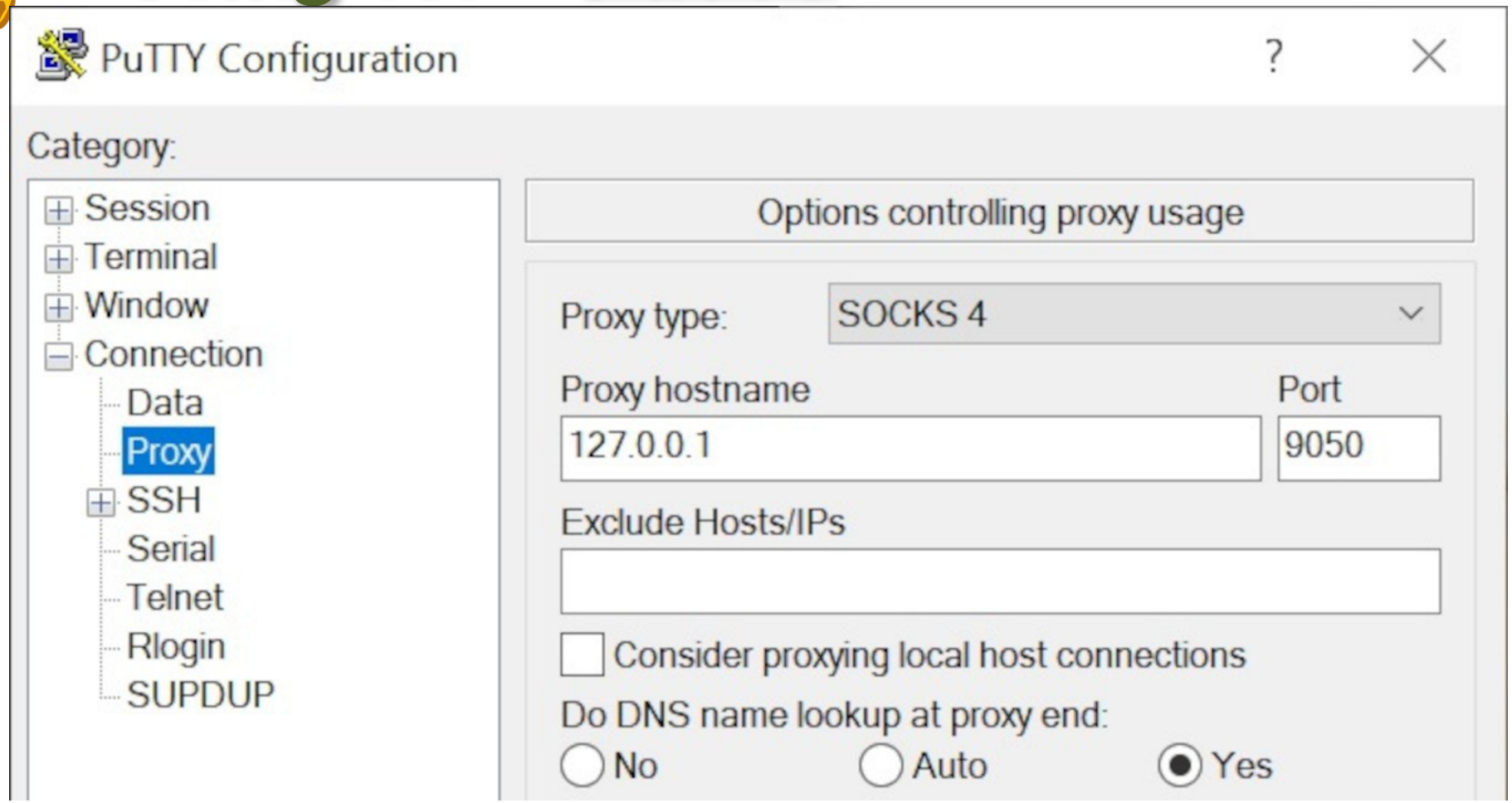


Using Tor – OpenSSH with Netcat

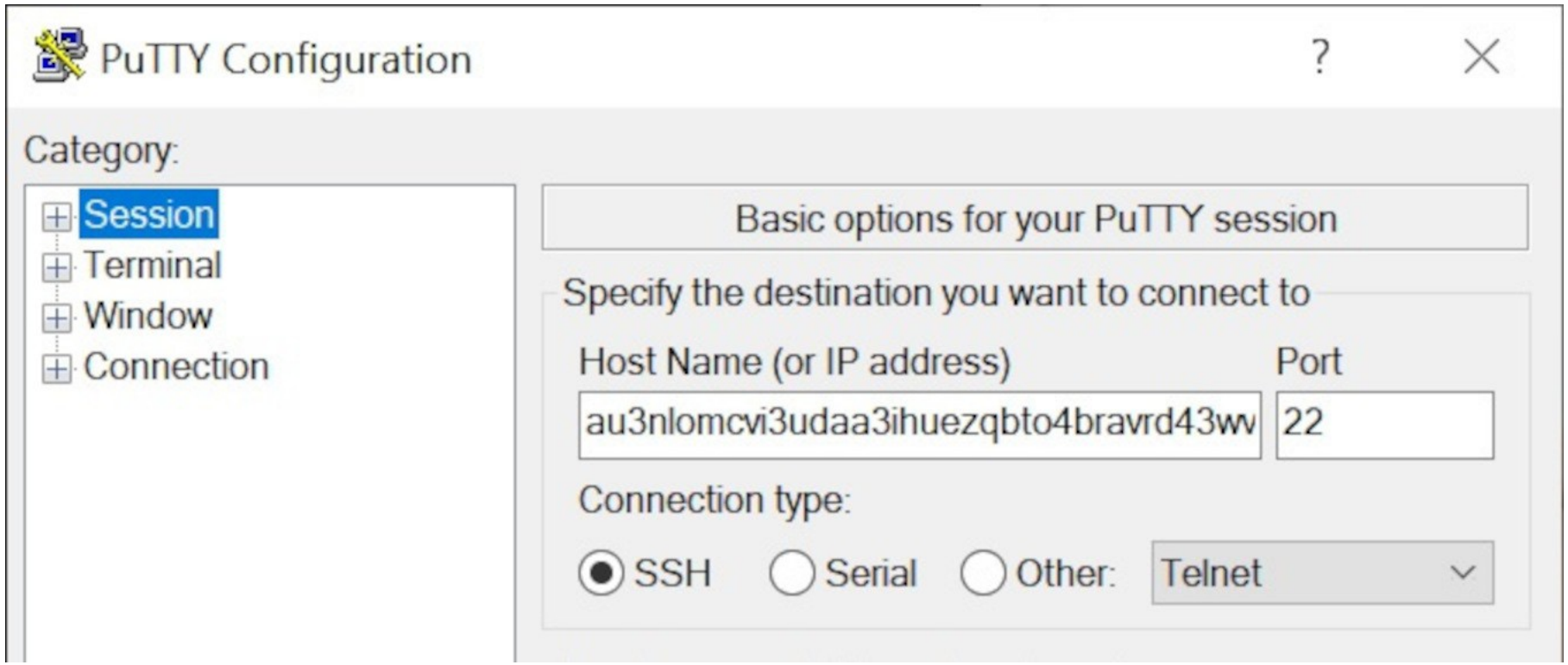

```
ssh -o \
ProxyCommand=\
'netcat -x 127.0.0.1:9050 %h %p' \
xxxxxxxx.onion
```

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

Using Tor – PuTTY



Using Tor – PuTTY



Using Tor – X3270

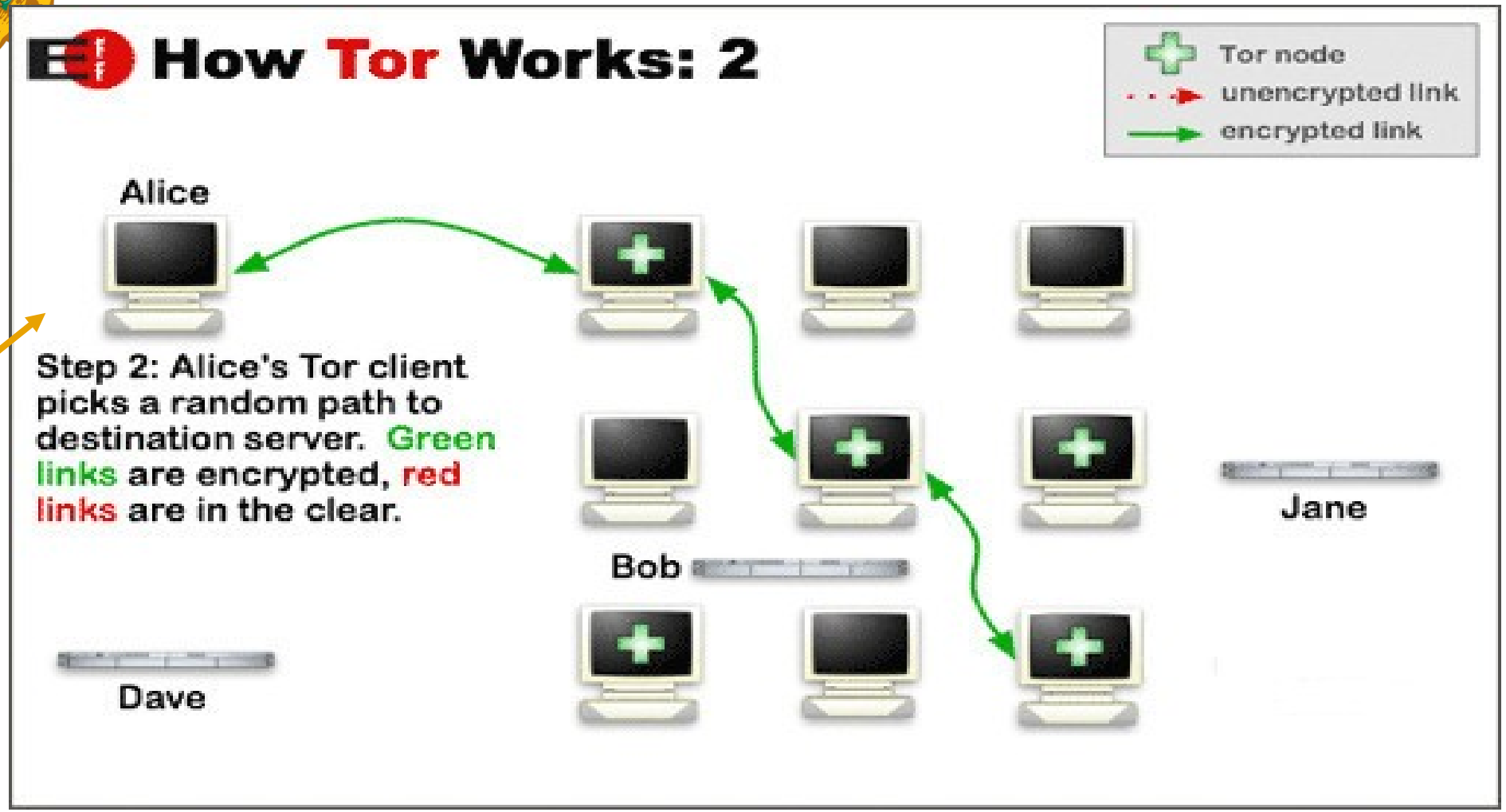

```
x3270 \
     -proxy socks4:127.0.0.1:9050 \
     xxxxxxxx.onion
```

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 (an IANA-blessed TLD)

Where's Bob?



Your Tor

No exit node

Does It Really Work?

Yup, some say so.

“Not Even the NSA Can Crack
the State Dept's Favorite Anonymous Network”
[Wikipedia, Foreign Policy, “The Cable”, wayback]

Is It Legal?

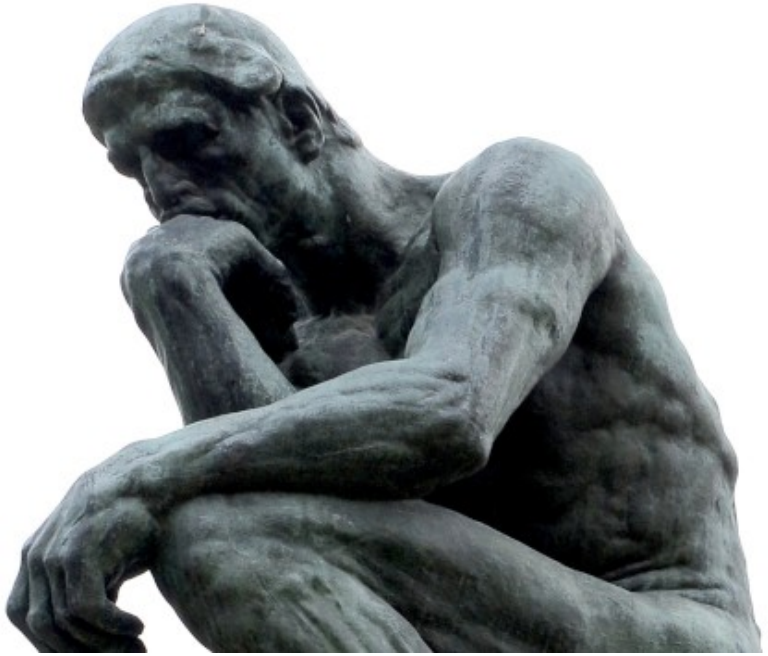
Absolutely, though it does get bad press.

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]

Benefit of using Tor

Tor gives you added control over your networking.



Effortless Risk-Free Reachability

Personal Example: Off-site Backup

- Raspberry Pi
- 6TB External (USB) Disk
- Wi-Fi or wired ethernet
- Tor Hidden Service

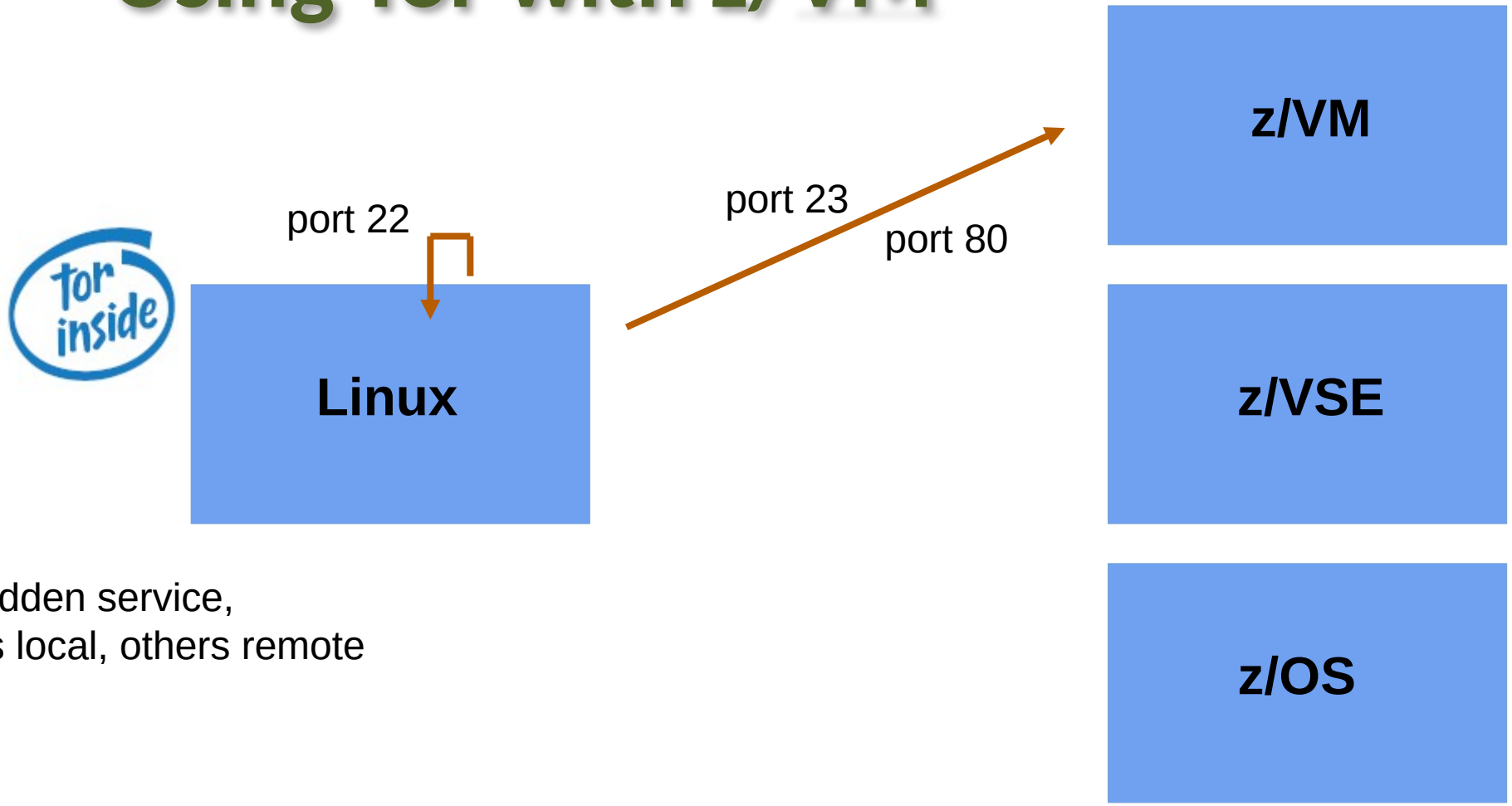

WD Elements

Using Tor with z/VM

"But Rick, what's this got to do with *VM*?"

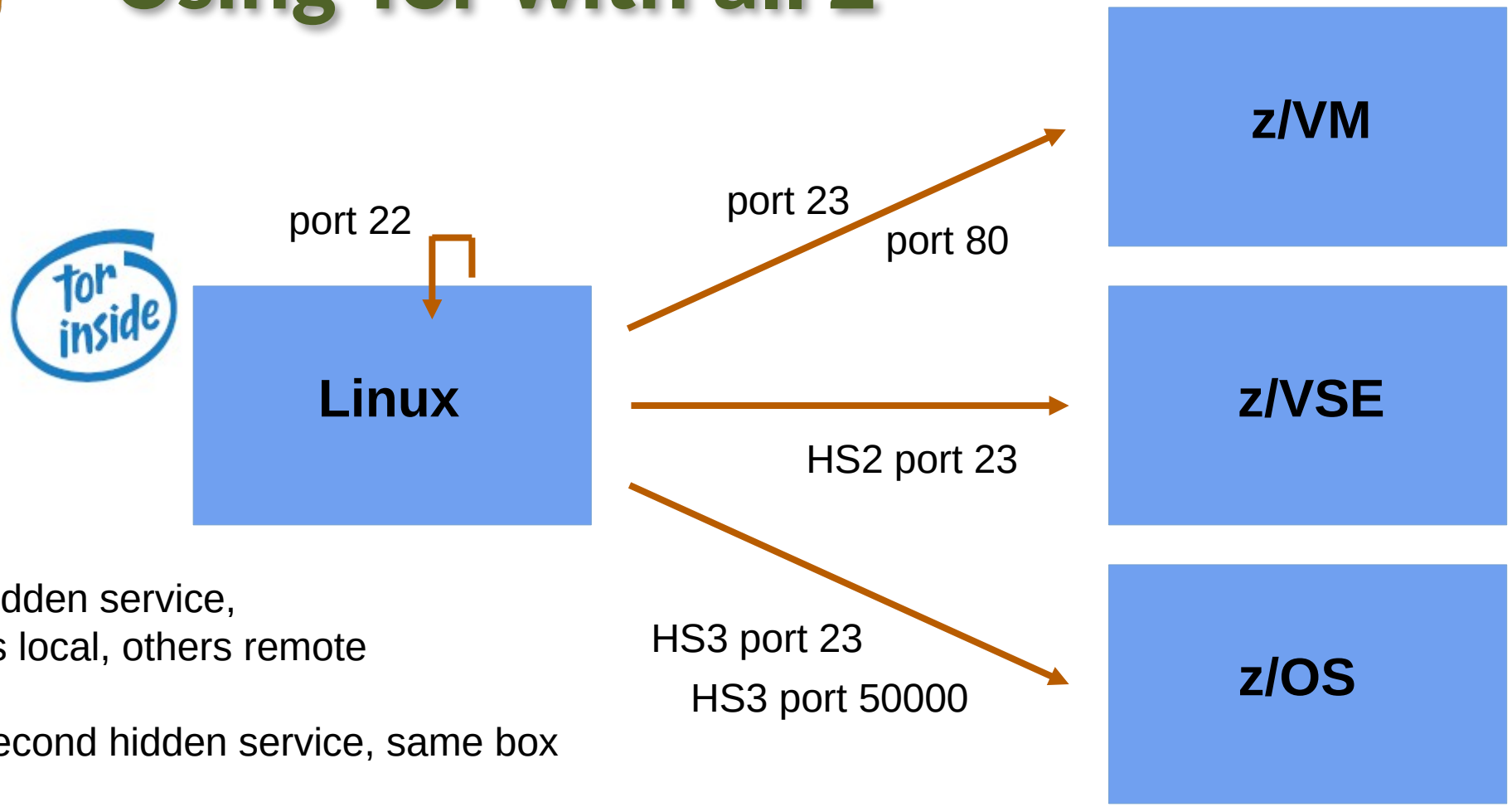
- VM (and VSE, MVS, TPF) is in the same DC (near subnet)
- Define "remote" (w/r/t the Tor host) hidden services
- Use it where PKI won't suffice; no conflict with PKI
- <u>No changes needed</u> to VM (nor VSE, TPF, MVS)

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`

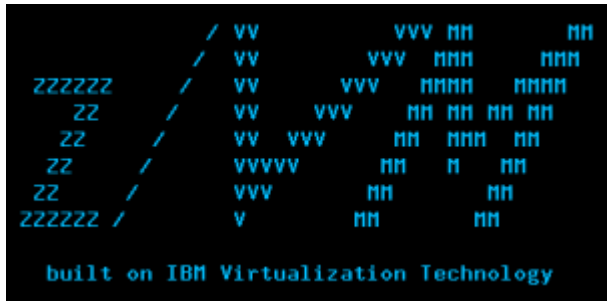
  `https://www.torproject.org/dist/tor-0.4.7.13.tar.gz.asc`

- Get it from your software package repository
  Debian, SUSE, RedHat and derivatives, BSD

Example RC file for Tor

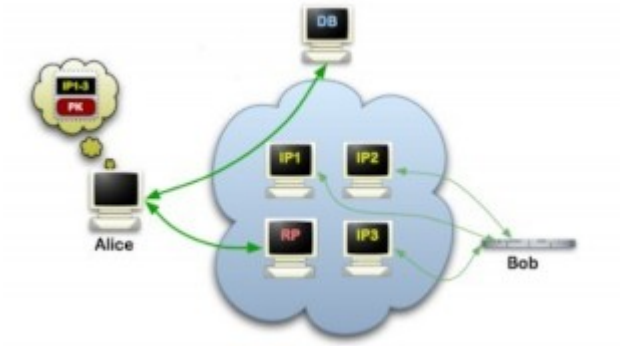
```
Nickname myzvmsystem
ContactInfo zVM Master <maint AT vm dot dom>
```

…



Example RC file with Hidden Service

```
…
HiddenServiceDir  /var/tor/hidden_service/
HiddenServicePort     22 127.0.0.1:22
HiddenServicePort     23 192.168.2.222:23
HiddenServicePort     80 192.168.2.222:80
```



Demo Time

.onion addresses (.onion hostnames)

The long and the short of it …

Originally: **`2hiyjpes6xu5ds7l.onion`**

Currently: **`au3nlomcvi3udaa3ihuezqbto4bravrd43wvehyhq24ricqkkwy2csyd.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?

We protect
the wrong things.

- 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?

It's all about trust!
It's all about control!

There is no Easy Button

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

The Pain of Certificate Management

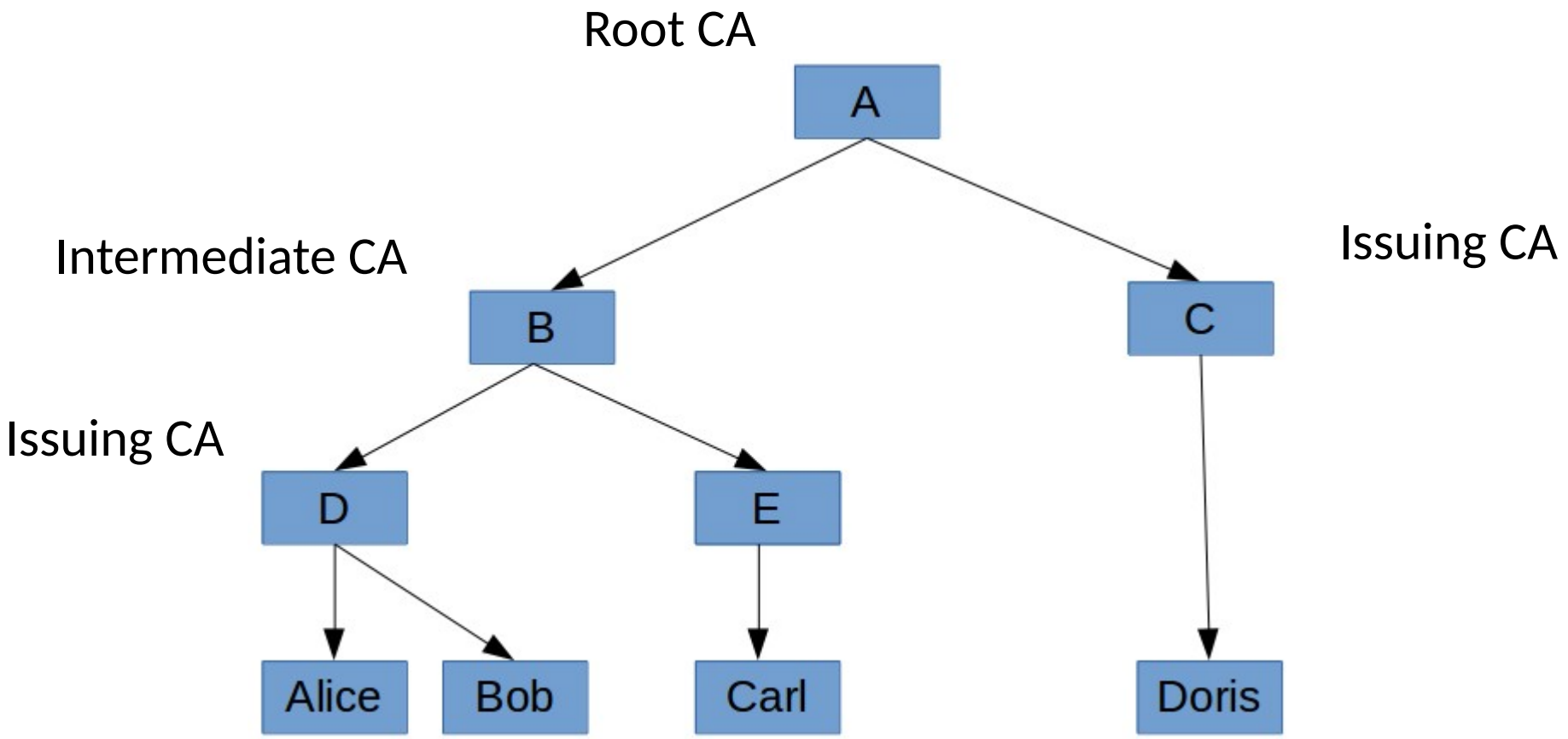
Comparing Tor with PKI

- 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)

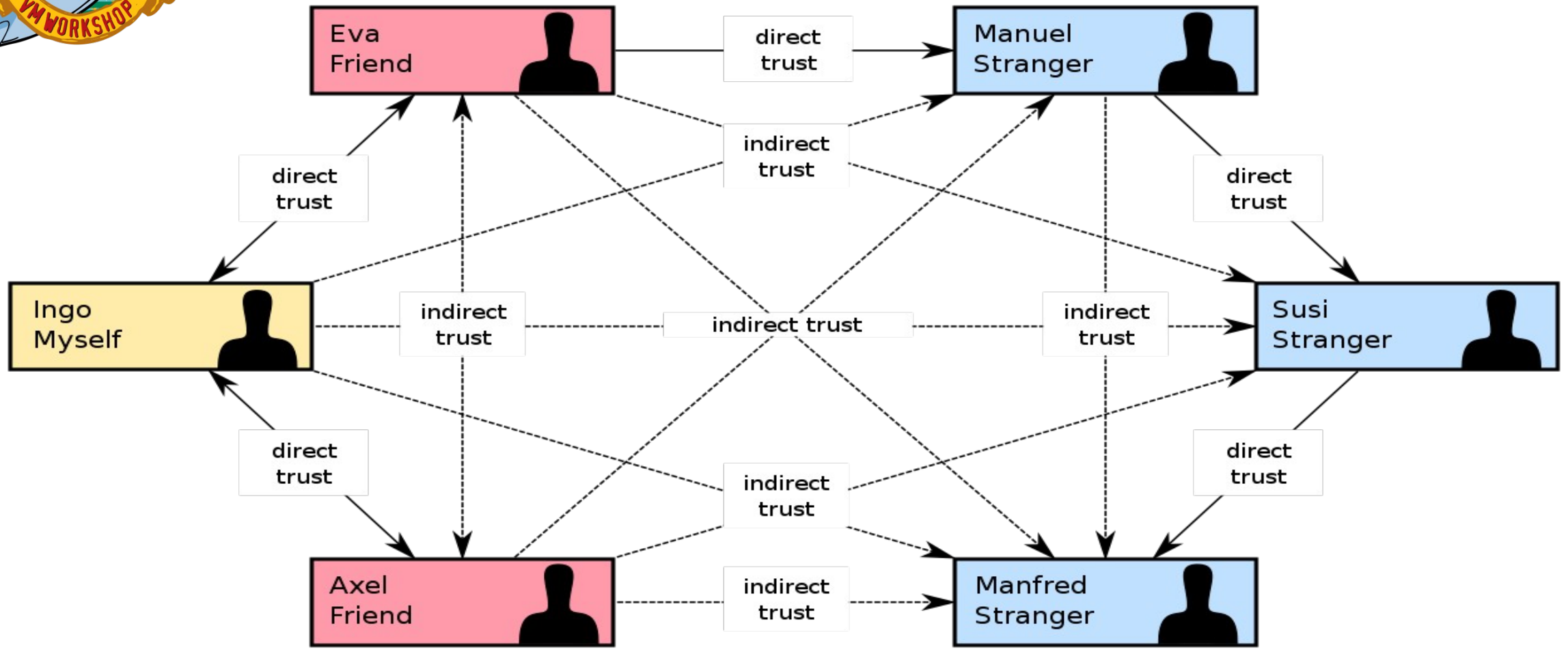
The Pain of Certificate Management

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



Trust Models



Conclusion … put Tor in the bag

And you should use Tor with z/VM and other Z systems

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

On the web and on the fob

Some files that might be of use …

- torforzvm/tor.init — traditional SysV INIT script
- torforzvm/tor.service — sample SystemD “service” file
- torforzvm/sshtor.sh — a wrapper shell script for 'ssh'
- torforzvm/torlinks.txt — various links

- cafortor/ — files for creating a PKI cert for HS

On the fob

Some packages that might be of use ...

- opt/tor-0.4.7.13/ what this talk is about
- opt/libevent-2.1.12/ in case you don't already have it
- opt/openssl-1.0.2u/ in case you don't already have it
- opt/gnupg-1.4.23/ everyone should have this
- opt/openssh-9.3p1/ everyone should have this
- opt/xmitmsgx-2.1.3/ a CMS 'XMITMSG' work-alike

Thank you!

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

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

http://www.casita.net/vmworkshop/2023/cafortor/

Thank you!

Or when you're "on" Tor …

[]=au3nlomcvi3udaa3ihuezqbto4bravrd43wvehyhq24ricqkkwy2csyd.onion

http://[]/vmworkshop/2023/torforzvm.pptx

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

http://[]/vmworkshop/2023/cafortor/

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`

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)

Securing z/VM and Linux using Tor Hidden Services

Rick Troth

2023 June 23, VM Workshop, Columbus

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!*"

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

The goal of *this* talk is to get people using Tor with z/VM, VSE, MVS, TPF, … all things Z.

Cybersec program at Cedarville University, I was allowed to present Tor, did not want the students to run afoul of university IT department.

Tor provides privacy (anonymization of your traffic) and some level of added security (difficult to spoof) so it makes sense to have in your arsenal as a VM admin.

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.

Some graphics courtesy of istockphoto.com.

Breadcrumbs can de-anonymize you.
Tor cannot stop you from “leaking”.

special disclaimer …

Many enterprises frown on things which are considered fringe. They consider Tor not suitable for corporate use. We will show some examples of “Tor hate” and how to respond to it.

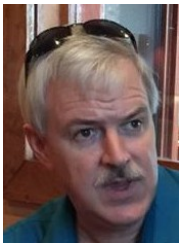
Tor is a powerful, yet easy-to-use tool.

When I gave a similar talk at for Cedarville University’s cybersec students, I warned them to not get sideways with the school’s IT department.
Same goes here.

Time was, many enterprises frowned on using Linux. They didn’t consider it suitable for corporate use.
Maybe the same goes for Tor and things like it.
Just sayin.

Tor is just a tool. It’s powerful, but it’s only a tool.

about:rick



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

"It's all about trust!"

BAE SYSTEMS

Voltage security

MICRO FOCUS

"It's all about trust" is one of my slogans, a recurring theme: trust is one reason for using open source. But even with source code, consider Thompson 1984.

https://www.cs.cmu.edu/~rdriley/487/papers/Thompson_1984_ReflectionsonTrustingTrust.pdf

I find myself increasingly concerned about security. Earned my CISSP while at Voltage.

BAE Systems, Inc. – a commanding breadth of capabilities

At BAE Systems, we serve, supply, and **protect those who protect us** in a company culture that is performance-driven and values-led. We aim to provide a vital advantage to our customers through our world-class defense capabilities across the air, land, maritime, and cyber domains, and by accelerating the pace of innovation. As one of the broadest and most geographically diverse international defense companies, we have an important role to play in contributing to the economic prosperity of the places where our people live and work, and in using our knowledge and technologies to reduce the environmental impacts of our operations.

Electronics

Land

Intelligence

Maritime

Support Services

We don’t have a cool logo.
We just make cool stuff.

Tunneling into Tor

What exactly is Tor?

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

What can Tor do for z/VM?

- Leverage Tor “Hidden Services” (HS) for z/VM TCP/IP
- Reachability without Risk

HS = “hidden service” throughout this presentation

We will not specifically cover *building* Tor from source, but slides discussing that are at the end of the presentation deck.

We will not discuss Tor startup (INIT or SystemD), but sample files are available at the web site.

My booth babe in the back has our biz cards
if you need a consultant to help.

Since we are at the VM Workshop,
the theme of this talk is “what can Tor do for z/VM?”.

about:security

- Privacy != Security and vice versa
- Anonymity != Privacy, but they do overlap
- Good algorithms are at the mercy of bad implementations
- Good implementations are at the mercy of bad deployment
- It’s all about trust

Tor provides two functions: encryption and routing. Tor is essentially a distributed VPN. It anonymizes your internet traffic (anonymizes your web surfing, but more than just web traffic) and yet works hand-in-hand with other mechanisms.

about:security

Security vs Usability, Closeted vs Networked + Hardened

- Don't react in fear; Consider the impact of the choices
- Compare with redundant systems, components, services

We don't have (and *should* have) redundant secure methods

"Security is a Process"

What is it about Tor that's interesting?
It makes anonymous, secure, private networking easy-to-do.

What is it about other methods that's frustrating?
There must be alternative access for when (normal) secured access fails. The Help Desk phone rings less often when there are reliable alternatives (and users know about them). But most other well-known solutions (especially, e.g., MFA) result in more risk of undesired/unintended lockout.

I hate the phrase "security is a process" but it's true. There is no fairy dust. There is no Easy button.

about:tor

The Onion Router

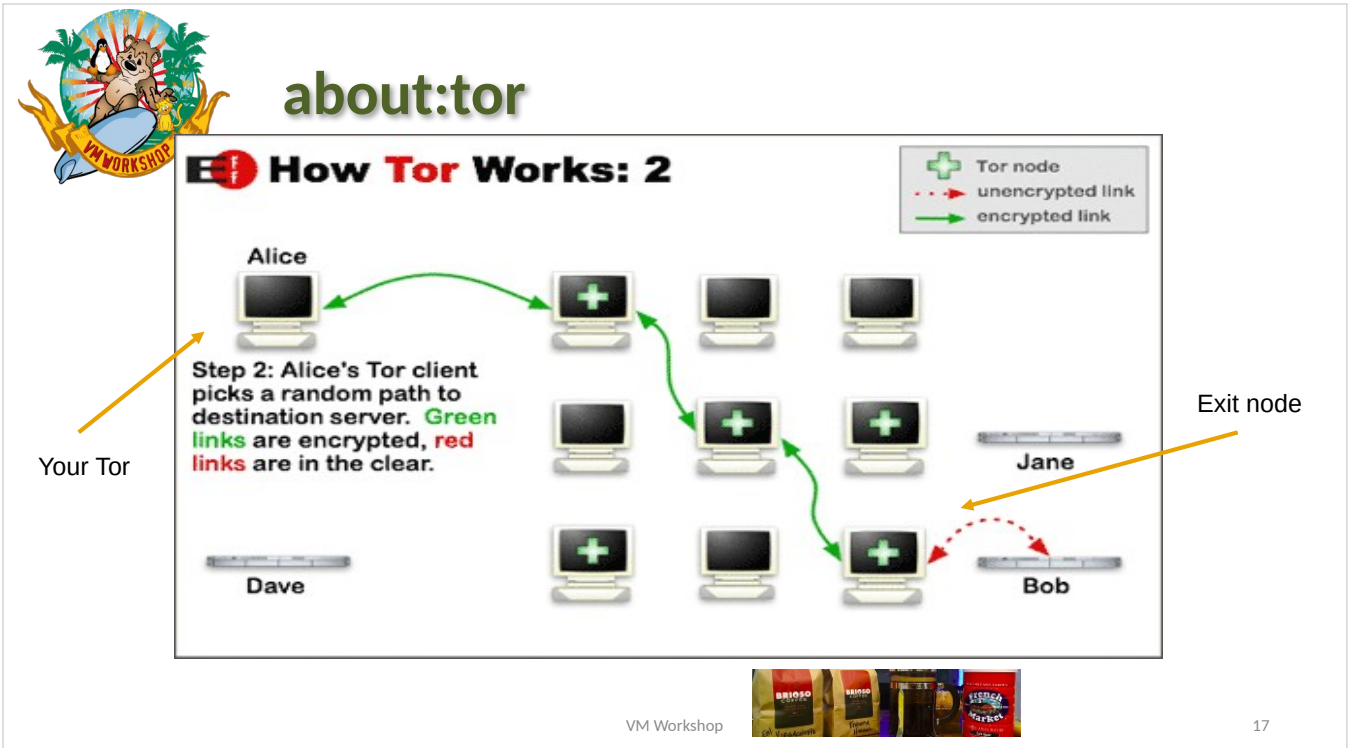
- http://www.torproject.org/
- Originally a US Navy project, first release 2002-September-20
- Other sponsors (e.g., EFF), now 501(c)(3)

"making the web safe for whistleblowers"

Tor provides encryption and routing.
Tor is a distributed VPN.

1995: engineers at the U.S. Naval Research Lab (NRL) suoght a way to make internet connections that don't reveal who is talking to whom. They wanted to protect Navy operatives in hostile waters. The result was some of the first designs and prototypes of what came to be called "onion routing".

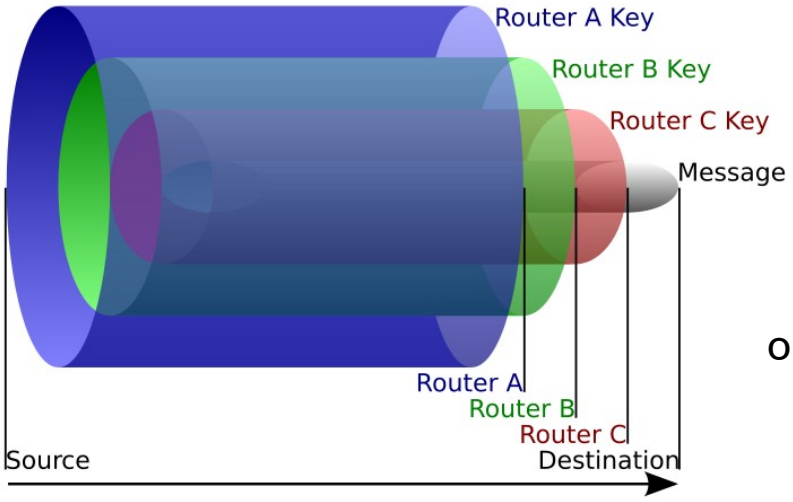
Early 2000s: Roger Dingledine continued the NRL work. The project came to be known as "Tor", an acronym for "The Onion Router".
Nick Mathewson joined soon after.



The entry point is local.
Your local ‘tor’ program connects with the fabric and provides a SOCKS proxy into that.

The last part, after the “exit node”, is visible to others. That part of your surfing can be profiled. (breadcrumbs again) But the exit node will vary, is used by others, and does not map back to you.

about:tor



onion routing

“Layers … ogres have layers.”

-- Shrek (Mike Meyers)

Using Tor

"But Rick, how do we *use* it?"

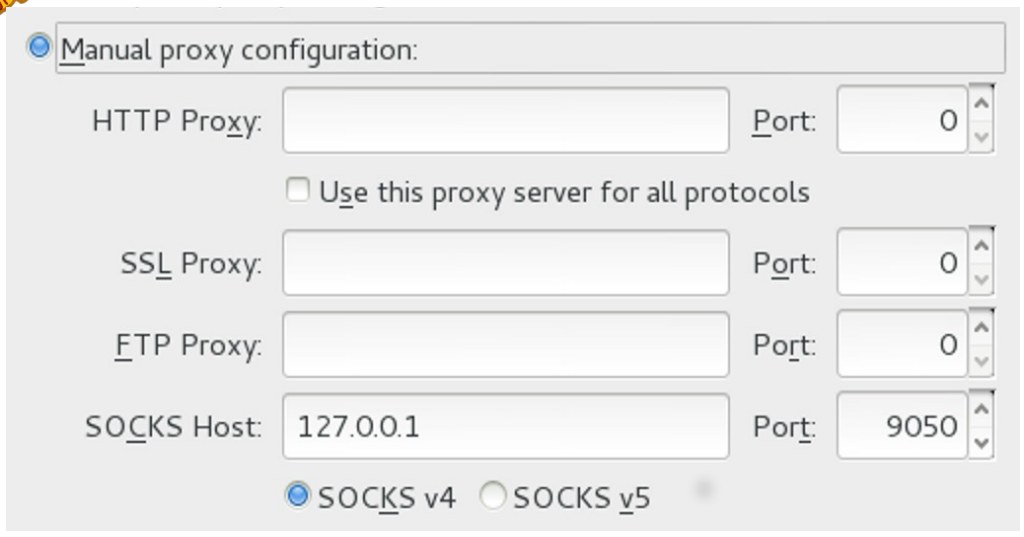
- Just run it
- *Don't* run it as root
- Use an RC file, perhaps **`/etc/tor/torrc`**
  else "not present, using reasonable defaults"
- State directory **`$HOME/.tor`** will be created
- Point at it as a SOCKS proxy, localhost port 9050

I use /etc/tor. It holds more than just the RC file.
I run 'tor' under a service account. That ID owns the "state" content, home directory /var/tor.

Note:
this mode of operation **does not** make you a target

The daemon can be started manually, via traditional Unix 'init' or via SystemD, or run as a co-process (e.g., to support the Tor Browser). It provides a SOCKS proxy by default.

Using Tor – SOCKS4a proxy



Uses “socks4” at local TCP port 9050.

The proxy port can be tunneled (e.g., over SSH). You don’t always have to run Tor locally just to use Tor as a client application.

Using Tor – avoid DNS leakage

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

Make sure that DNS requests are handled via Tor. Running DNS through the proxy is not always the default, so you have to check it manually.

Using Tor – OpenSSH with 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.

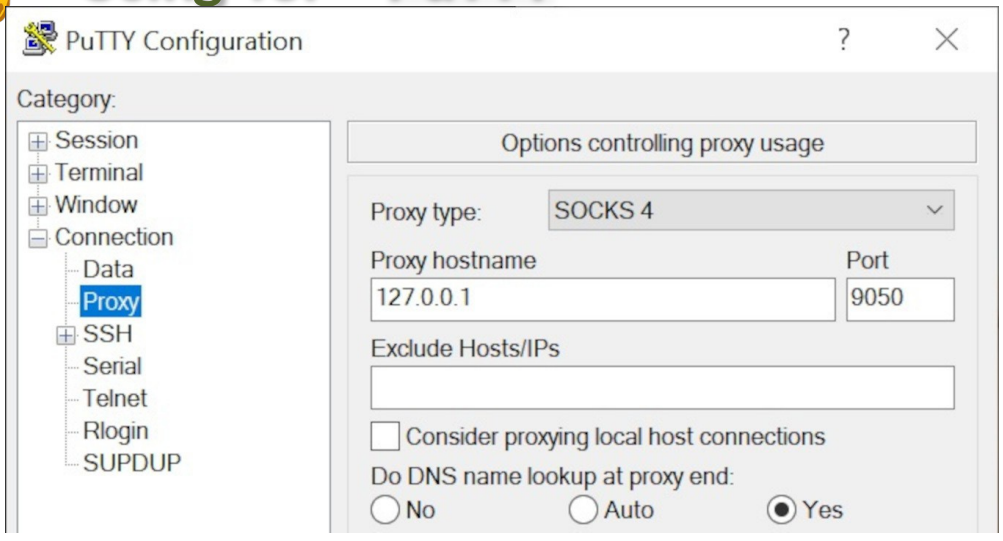
Important: Tor is not limited to web pages.
ANY TCP traffic can be relayed through Tor.

OpenSSH does not have built-in proxy support, but 'netcat' does and SSH can punt to 'netcat'.

This extends to Linux "console server" too.

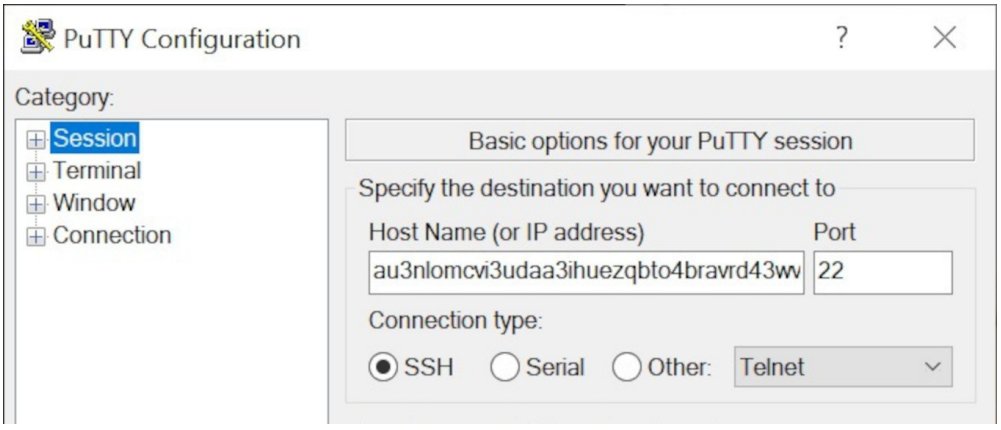
Same "ProxyCommand" option works for 'scp' too.

Using Tor – PuTTY



Again, force DNS through the proxy.

Using Tor – PuTTY



The filled-in hostname is a “.onion” hostname. More on that in following slides.

Using Tor – X3270


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


X3270 has built-in proxy support. Yay!

What's with the "dot onion"?

Introducing … *hidden services* [the crowd cheers]

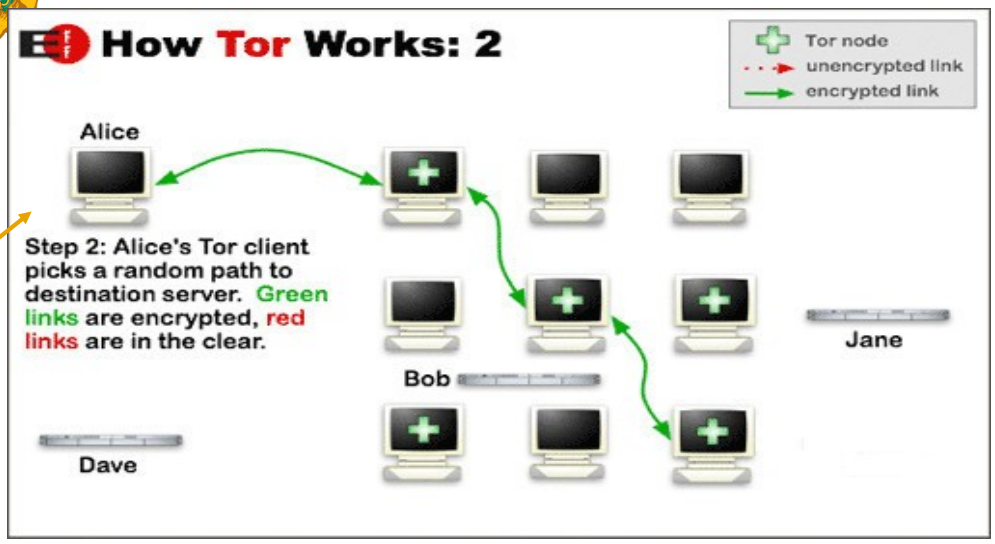
- Traffic past an "exit node" is visible outside
- Traffic handled by a "hidden service" is not visible
- Hidden services are known by "**`.onion`**" hostnames (an IANA-blessed TLD)

Hidden services are thus named because they are not visible (indeed, not even reachable) from the public internet.

The hostname of a hidden service must not resolve in DNS, so Tor developers chose ".onion" suffix.

Lately the name is "onion services".
Originally, hidden services was a labor of love for the Tor development team. Now it has become a significant feature.

Where's Bob?



Your Tor

No exit node

No red links here! (Nothing outside Tor network.)
Bob is not accessed via an exit node.
Bob is hiding somewhere in the fabric.
There's a private key (automatically generated) and a .onion hostname (derived from the key), but Bob's location is unknown.

Whoever maintains Bob can relocate him (the key and related state files) to another physical host should the need arise.

https://2019.www.torproject.org/docs/onion-services

Does It Really Work?

Yup, some say so.

“Not Even the NSA Can Crack
the State Dept's Favorite Anonymous Network”
[Wikipedia, Foreign Policy, “The Cable”, wayback]

I got this from footnotes in the Wikipedia page about Tor.

How is it that people trust SELinux from the NSA and yet they don't trust Tor from the US Navy?

Is It Legal?

Absolutely, 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”.

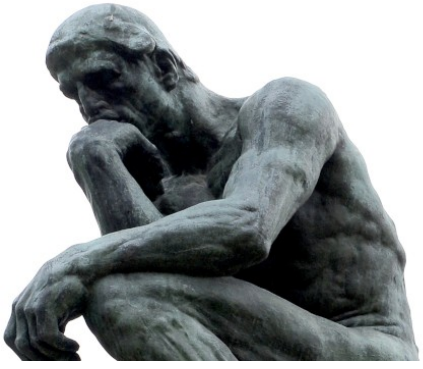
[Wikipedia, UC Berkeley, wayback]

Tor is attacked by good guys to stop the bad guys.
Tor is attacked by bad guys to stop the good guys.
Tor itself is just a tool. Your laptop is just a tool.
Is your laptop legal?

If you were not a fan of the Clipper Chip
then you might be interested in using Tor.

Benefit of using Tor

Tor gives you added control over your networking.



Tor is also a potential threat to people and organizations which need (or just want) ultimate control over end-user networking. Tor puts you back in control.

If you were not a fan of the Clipper Chip then you might be interested in using Tor.

Effortless Risk-Free Reachability

Personal Example: Off-site Backup

- Raspberry Pi
- 6TB External (USB) Disk
- Wi-Fi or wired ethernet
- Tor Hidden Service

This is the backup server at our cabin.
Shown is the Raspberry Pi
coupled with velcro to its DASD farm.

The Tor network may be the only non-commercial VPN presently available (to the general public). One *can* establish a personal CA and use certs with (e.g.) OpenVPN, but that involves more work than Tor HS.

Identical backup server at my son’s apartment.

Using Tor with z/VM

"But Rick, what's this got to do with *VM*?"

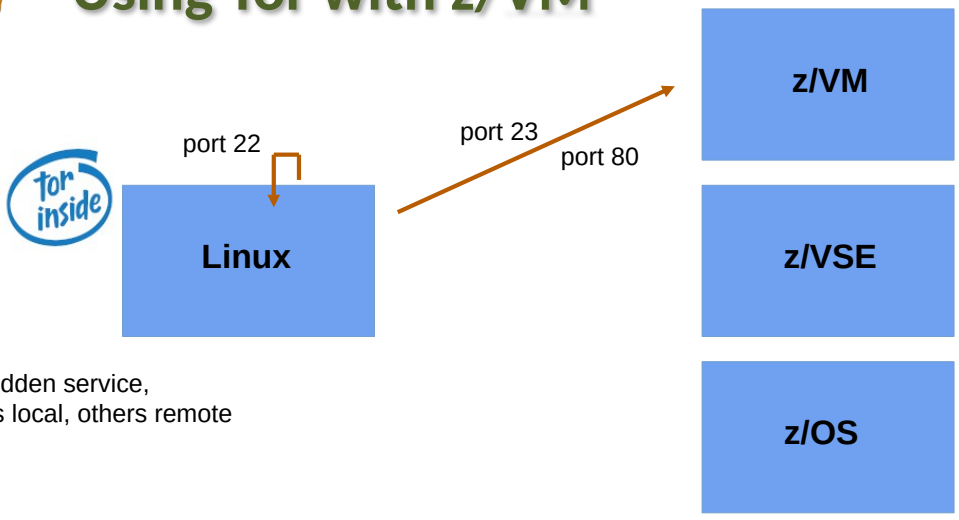
- VM (and VSE, MVS, TPF) is in the same DC (near subnet)
- Define "remote" (w/r/t the Tor host) hidden services
- Use it where PKI won't suffice; no conflict with PKI
- No changes needed to VM (nor VSE, TPF, MVS)

VM is a near neighbor (same DC) to the Linux Tor host
Define hidden services from Linux directed at VM
You can define HS to *any* near neighbors,
thus VSE, MVS, TPF, other Linux, … anything

No changes needed to CP, CMS, MVS, TSO, etc.

There is no conflict between PKI (SSL/TLS) and Tor. You can run Tor without losing SSL service. The trust models between the two are the biggest operational difference. Tor is also stronger w/r/t privacy. (HS are very very very difficult to trace and re-identify.)

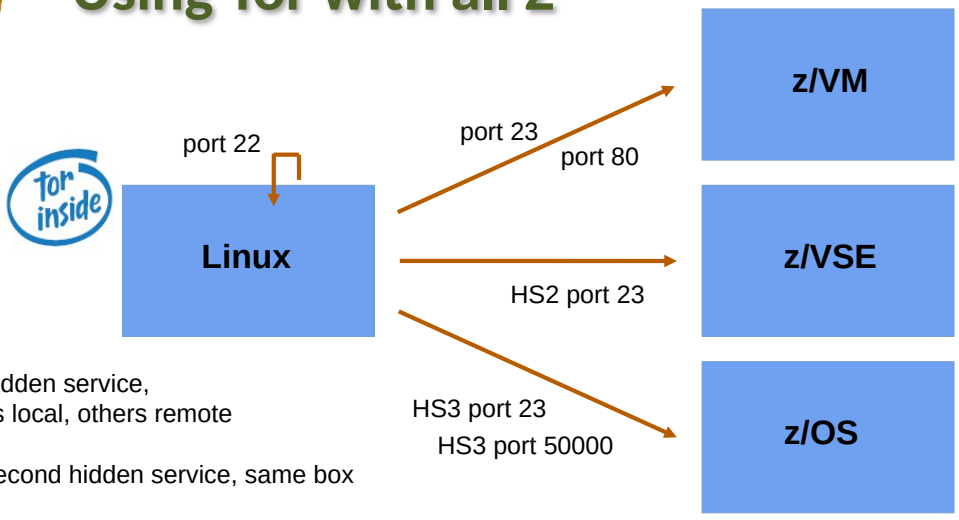
Using Tor with z/VM



Define a hidden service,
some ports local, others remote

This slide shows arbitrary TCP port choices.
You can assign an incoming Tor HS port
to any address:port pair you need.

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

You can define any number of Tor hidden services. Consider one for VM and Linux, another for VSE, another for MVS (here with DB2).

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
  Debian, SUSE, RedHat and derivatives, BSD

More details on compiling Tor yourself at end of this presentation.

Debian is special here:
it can *use* Tor for system updates.
Install the “apt-transport-tor” package.

Example RC file for Tor

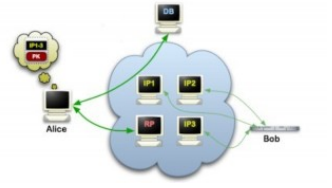
```
Nickname myzvmsystem
ContactInfo zVM Master <maint AT vm dot dom>
```

…

When supporting z/VM,
Tor would actually run on a Linux guest,
not on CMS.

Example RC file with Hidden Service

```
…
HiddenServiceDir  /var/tor/hidden_service/
HiddenServicePort    22 127.0.0.1:22
HiddenServicePort    23 192.168.2.222:23
HiddenServicePort    80 192.168.2.222:80
```



Here we define a hidden service.
A key pair will be generated by Tor, if such does not exist already. A .onion hostname will be derived. The files will be placed into the indicated directory.
There can be any number of hidden services (HS) defined, a separate directory for each.
The address side of the address:port pair can be any reachable address, not just the local host (where Tor is running).

Demo Time

.onion addresses (.onion hostnames)

The long and the short of it …

Originally: **2hiyjpes6xu5ds7l.onion**

Currently: **au3nlomcvi3udaa3**
ihuezqbto4bravrd
43wvehyhq24ricqk
kwy2csyd.onion

Hidden service addresses originally had
16 characters to the left of “.onion”.
Current hidden service addresses have
56 characters to the left of “.onion”.
This is ostensibly to make them more secure.

These are actually “v2” and “v3”, so there was presumably a “v1” back in the day, before my time.

These hostnames, and their counterpart keys, are derived by Tor when the given hidden service is first instantiated.

Popular .onion Sites

- Protonmail
- Keybase
- Debian
- DuckDuckGo
- Facebook

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

For Debian, install the “apt-transport-tor” package.

A list of long .onion hostnames is in a file alongside the download of this presentation.

Q: Why use a hidden service
if the site itself is public?
A: Hides that “last hop” from tracking.
It doesn’t help hide the site, but may help the users.

https://en.wikipedia.org/wiki/
List_of_Tor_onion_services

The Pain of Certificate Management

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

We protect
the wrong things.

- Come back next year, do it all over again

Tor is not better than PKI, but for some things it is easier. Tor has the objective advantage of obscuring traffic patters (visible when using plain SSL/TLS).

The Pain of Certificate Management

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

It's all about trust!
It's all about control!

There is no Easy Button

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

This year (2023), GitHub.com changed their SSH server key. I heard about it on IBM-MAIN or would have been caught quite by surprise and likely panicked. (not PKI but you get the idea)

More recently, IBM switched CAs and notified customers about the change. Probably not strictly required for web browsing (since root certs for both CAs were likely already in browser trust store), but necessary notification for customers using software verification via PKI.

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)

There must always be a trust anchor.
Hidden service “certs” (keys) are created automatically.
Unlike SSL/TLS, the traffic (connections) is hidden too.

For some things, Tor is easier than TLS/SSL PKI.
In some ways, it’s just different.
With respect to traffic analysis, it’s better.

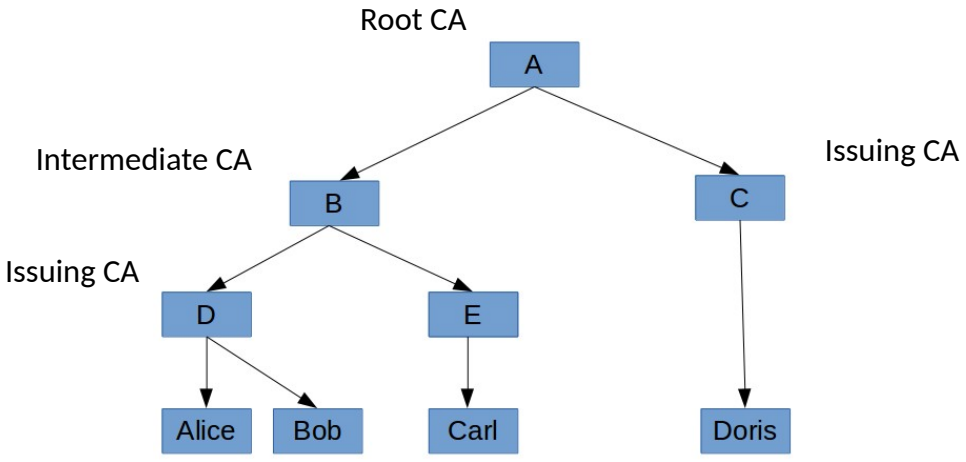
The Pain of Certificate Management

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 …

Who ya gonna trust?

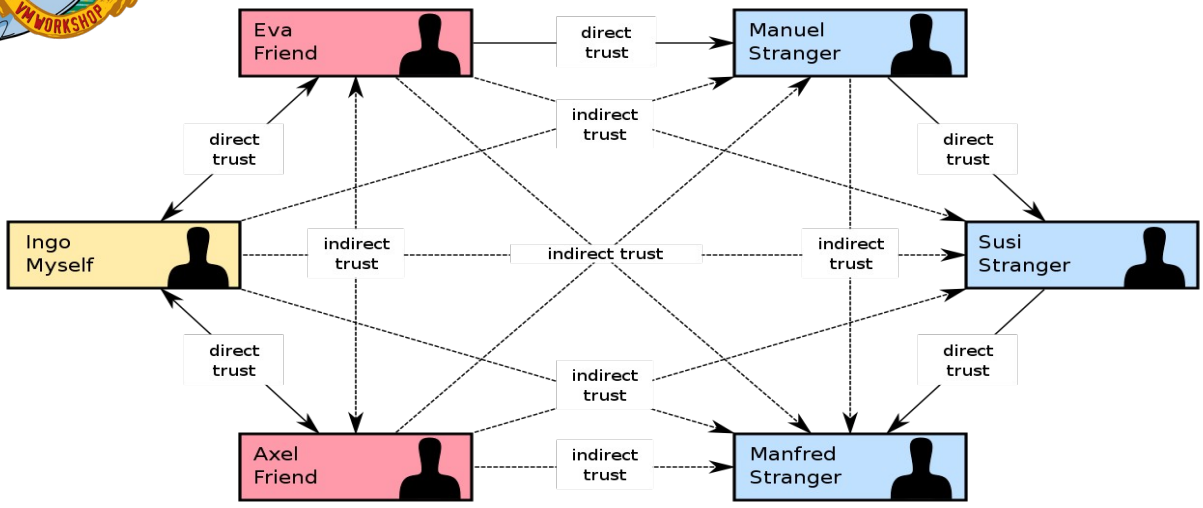
The PKI system was designed to generate business for the Certificate Authorities (CAs). With certificates needing regular renewal, the CAs have an ongoing stream of repeat customers. It’s nice to have residual income.

Trust Models



This is the PKI model.

Trust Models



This is the PGP model.

Conclusion … put Tor in the bag

And you should use Tor with z/VM and other Z systems

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

If you were not a fan of the clipper chip,
then you should consider Tor.

On the web and on the fob

Some files that might be of use …

- torforzvm/tor.init — traditional SysV INIT script
- torforzvm/tor.service — sample SystemD “service” file
- torforzvm/sshtor.sh — a wrapper shell script for 'ssh'
- torforzvm/torlinks.txt — various links

- cafortor/ — files for creating a PKI cert for HS

On the fob

Some packages that might be of use ...

- opt/tor-0.4.7.13/ what this talk is about
- opt/libevent-2.1.12/ in case you don't already have it
- opt/openssl-1.0.2u/ in case you don't already have it
- opt/gnupg-1.4.23/ everyone should have this
- opt/openssh-9.3p1/ everyone should have this
- opt/xmitmsgx-2.1.3/ a CMS 'XMITMSG' work-alike

There is a ‘setup’ script in each package directory which will create sym-links following the Chicory method.

The package directories each also contain wrapper logic for ‘make so that you can re-build any of these yourself. Remove “arc” sym-link or create the path it points to.

Thank you!

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

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

http://www.casita.net/vmworkshop/2023/cafortor/

Thank you!

Or when you're "on" Tor ...

[]=au3nlomcvi3udaa3ihuezqbto4bravrd43wvehyhq24ricqkkwy2csyd.onion

http://[]/vmworkshop/2023/torforzvm.pptx

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

http://[]/vmworkshop/2023/cafortor/

Put that 62-character address in place of the square brackets. Works if you're "on" Tor.

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 9ec002felc9ca517 **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)