Hidden services for newbies

Erik Knechtel
Tor HQ, 20-March-2017

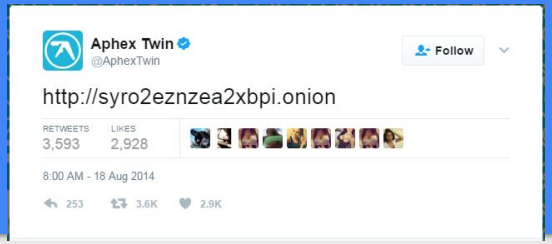
Goal of this talk

- Why run a hidden service?
- Famous .onions
- How to set one up
- .com vs .onion
- Not going to cover: What is Tor? How to host a website?

Why create a .onion?

- Allow people in oppressive countries to visit
- Allow people to visit while hiding their location/IP from your ISP, as well as from you
- Take pressure off the exit nodes
  - Why do the exit nodes need help? Answer: Tough legal position.
- Ensure strong security at every point
  - Do you need https?
- $0.00
- Debian/GNU/FLOSS repos should have .onions – encryption export laws, targeting of linux users here and abroad (NSA: Linuxjournal.com visitors = terrorists)
- “NAT punching”

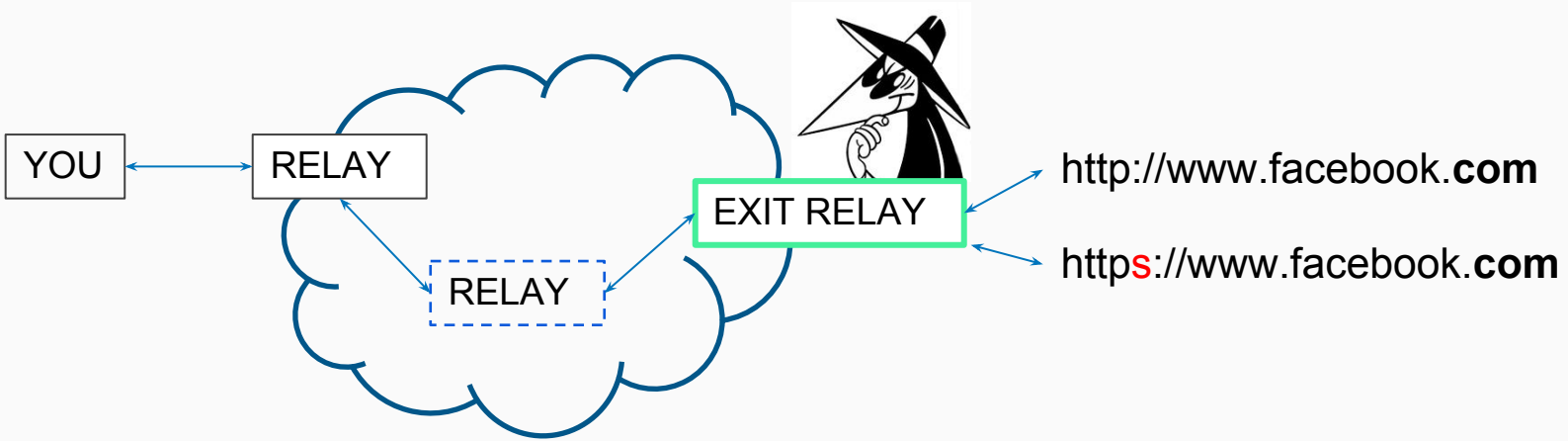
.onions in the news



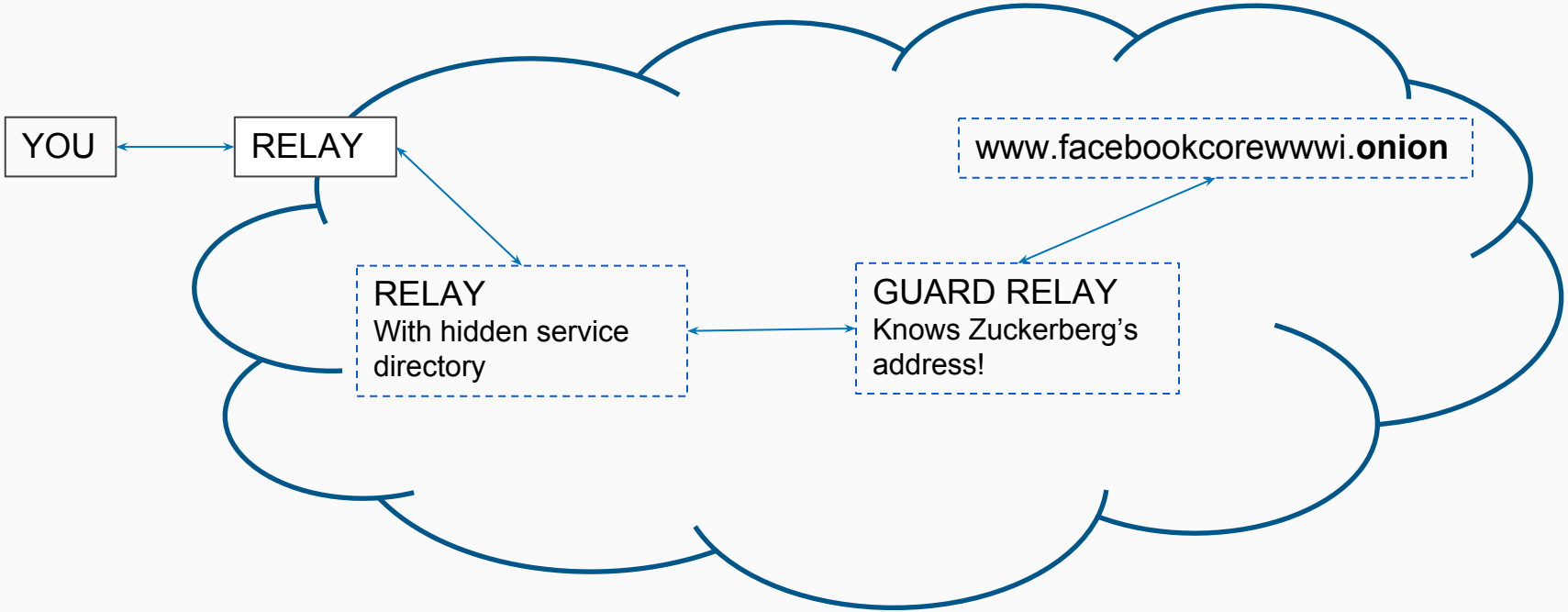
Aphex twin released an album on a .onion address in 2014. Celebrity promotion of anti-surveillance tech! (Do all .onions look like Aphex Twin song names or is that just me?)

Facebook, in April 2016: “In June 2015, over a typical 30 day period, about 525,000 people would access Facebook over Tor e.g.: by using Tor Browser to access www.facebook.com or the Facebook Onion site, or by using Orbot on Android. This number has grown – roughly linearly – and this month, for the first time, we saw this “30 day” figure **exceed 1 million people**.“

Clear web vs. Onionland



Clear web vs. Onionland



How to set up a .onion webpage

- Get the Tor service running. For example:
  - `$> sudo apt-get install tor; sudo service tor start`
- Get a webpage hosted: Apache, Nginx, Caddy, ... hosting .html, .jpg, etc on localhost:port
  - For example: a website being served on 127.0.0.1:80
- Edit the .torrc file to have Tor capture the web hosting service: port 80 → 9001

```
############### This section is just for location-hidden services ###

## Once you have configured a hidden service, you can look at the
## contents of the file ".../hidden_service/hostname" for the address
## to tell people.
##
## HiddenServicePort x y:z says to redirect requests on port x to the
## address y:z.

#HiddenServiceDir /var/lib/tor/hidden_service/
#HiddenServicePort 80 127.0.0.1:80

#HiddenServiceDir /var/lib/tor/other_hidden_service/
#HiddenServicePort 80 127.0.0.1:80
#HiddenServicePort 22 127.0.0.1:22
```

HiddenServiceDir has your private key and hostname.

Hostname = .onion address = summary of public key.

Setup continued

- Tor sets up a circuit to advertise that .onion and route requests
- A relay or relays are chosen to know your service's "true" location. All traffic comes through them.
    - Adversaries may DDOS those relays to try and make Tor choose one of theirs. Using relays you control as entry points to your service provides a canary in the coal mine.

My experience

- Started running a bridge relay after the Turkish Gezi protests in spring 2013
- Made a personal website, www.knek-tek.me, decided to make a .onion version
- Only used Tor Project's hidden service FAQs/tutorials and Apache tutorials
- Followed example of existing Beaglebone webpage for hosting a site
- Used scallion to create vanity domain (https://github.com/lachesis/scallion)
- www.knek-tek.me → http://knektek7naqk2234.onion
- www.facebook.com → https://www.facebookcorewwwi.onion (holy GPU cycles batman!)

.com vs .onion

- The clear web = The commercial web
  - Registrars, hosting, https certs, whois privacy fee, annual renewal, … = $$$
- .onion is FLOSS/OSHW all the way if you have the hardware
- Twitter Mirai attack: If they had had a http://twitterxxxxxxxxx.onion, they wouldn't have gone down. Clearweb DNS != Tor DNS. Note that using tor to connect to twitter.com still wouldn't work. Only .onion!

Questions?

Shoutout to Tech Solidarity Meetups: @TechSolidarity, @Pinboard

Contact: maciej@ceglowski.com, or Signal: +1 415 610 0231