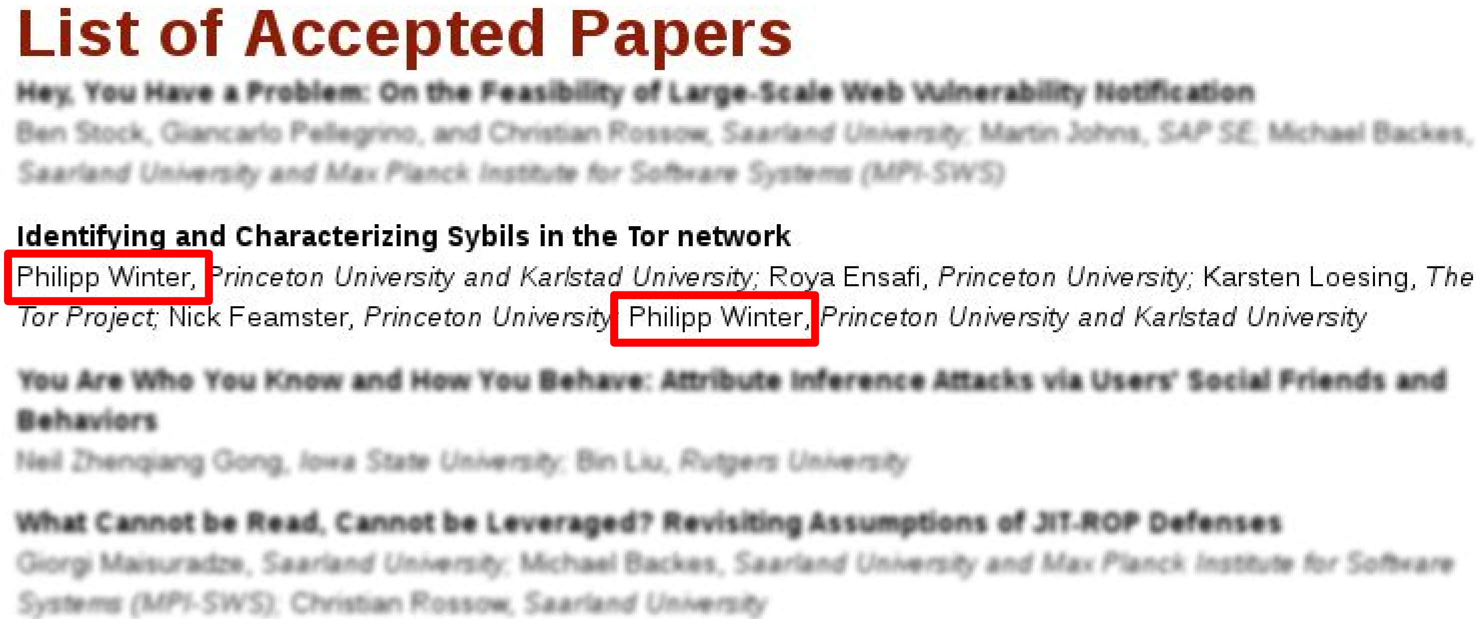
Identifying and characterizing Sybils in the Tor network

August 12, 2016
USENIX Security Symposium

Philipp Winter	Princeton University and Karlstad University
Roya Ensafi	Princeton University
Karsten Loesing	The Tor Project
Nick Feamster	Princeton University



List of Accepted Papers

Hey, You Have a Problem: On the Feasibility of Large-Scale Web Vulnerability Notification

Ben Stock, Giancarlo Pellegrino, and Christian Rossow, *Saarland University;* Martin Johns, *SAP SE;* Michael Backes, *Saarland University and Max Planck Institute for Software Systems (MPI-SWS)*

Identifying and Characterizing Sybils in the Tor network

Philipp Winter, *Princeton University and Karlstad University;* Roya Ensafi, *Princeton University;* Karsten Loesing, *The Tor Project;* Nick Feamster, *Princeton University;* Philipp Winter, *Princeton University and Karlstad University*

You Are Who You Know and How You Behave: Attribute Inference Attacks via Users' Social Friends and Behaviors

Neil Zhenqiang Gong, *Iowa State University;* Bin Liu, *Rutgers University*

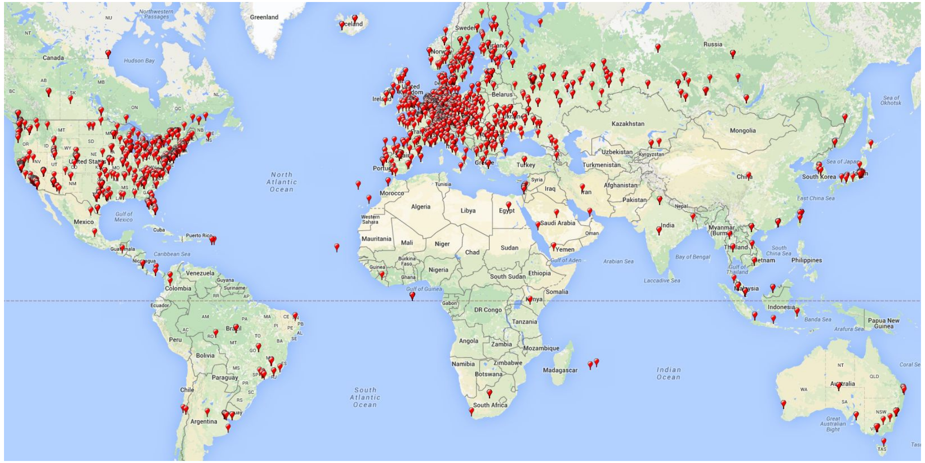
What Cannot be Read, Cannot be Leveraged? Revisiting Assumptions of JIT-ROP Defenses

Giorgi Maisuradze, *Saarland University;* Michael Backes, *Saarland University and Max Planck Institute for Software Systems (MPI-SWS);* Christian Rossow, *Saarland University*

The double-edged sword of volunteer-run networks

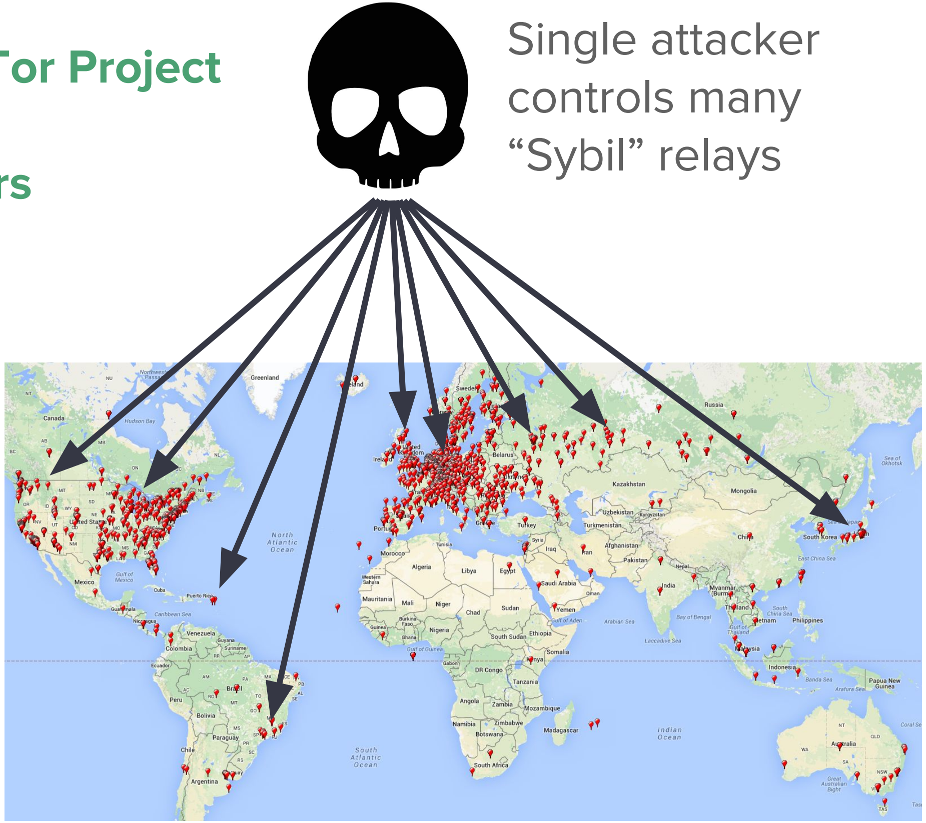
- The Tor **code** is developed by **The Tor Project**
- The Tor **network** is run by **volunteers**
- Currently ~7,000 relays
- **Low** barrier of entry

Tor relays as of Aug 2016

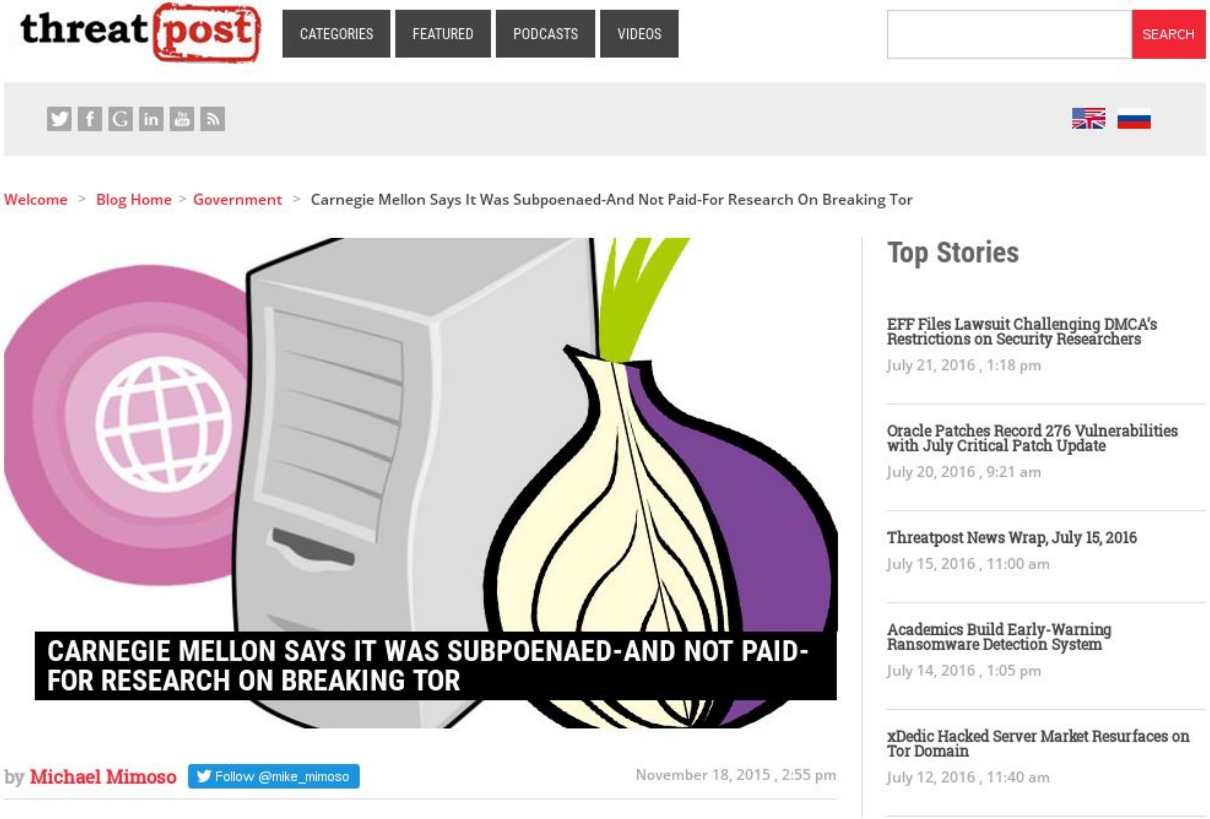


The double-edged sword of volunteer-run networks

- The Tor **code** is developed by **The Tor Project**
- The Tor **network** is run by **volunteers**
- Currently ~7,000 relays
- **Low** barrier of entry



Single attacker controls many “Sybil” relays



Welcome > Blog Home > Government > Carnegie Mellon Says It Was Subpoenaed-And Not Paid-For Research On Breaking Tor

CARNEGIE MELLON SAYS IT WAS SUBPOENAED-AND NOT PAID-FOR RESEARCH ON BREAKING TOR

by Michael Mimoso

November 18, 2015 , 2:55 pm

Top Stories

EFF Files Lawsuit Challenging DMCA's Restrictions on Security Researchers

July 21, 2016 , 1:18 pm

Oracle Patches Record 276 Vulnerabilities with July Critical Patch Update

July 20, 2016 , 9:21 am

Threatpost News Wrap, July 15, 2016

July 15, 2016 , 11:00 am

Academics Build Early-Warning Ransomware Detection System

July 14, 2016 , 1:05 pm

xDedic Hacked Server Market Resurfaces on Tor Domain

July 12, 2016 , 11:40 am

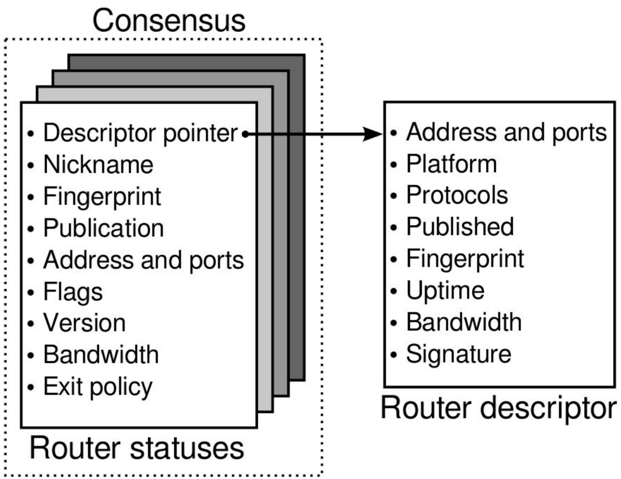
Existing Sybil defenses don’t help

- Social network-based defenses **don’t apply**
- Proof-of-work-based defenses **inherent** to running a relay
- Instead, we leverage two observations to **detect** Sybils
  - Sybils often **controlled** similarly
  - Sybils often **configured** similarly

```
Nickname IP address     ORPort DirPort Flags                                   Version     OS      Bandwidth
Unnamed  204.45.15.234  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.15.235  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.15.236  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.15.237  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.250.10  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.250.11  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.250.12  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.250.13  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
Unnamed  204.45.250.14  9001   9030    Fast|Guard|HSDir|Stable|Running|Valid|V2Dir 0.2.4.18-rc FreeBSD 26214400
```

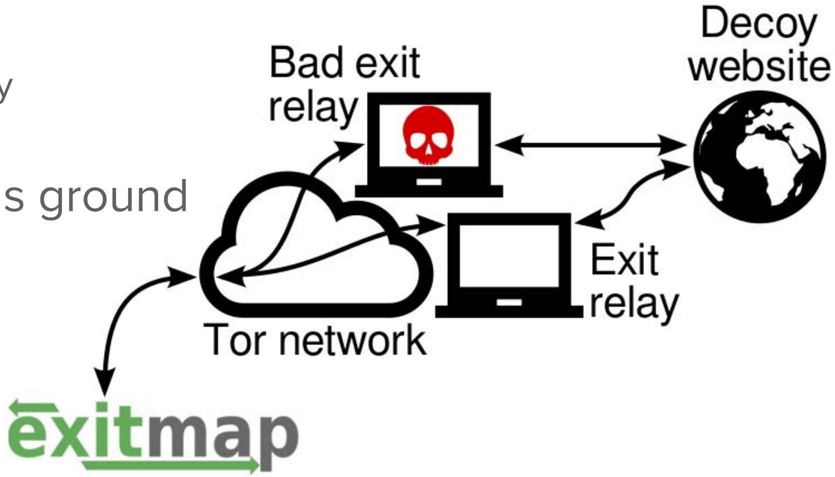
Passive dataset

- The Tor Project **archives** lots of data
  - Available at collector.torproject.org
- **Network consensus** hourly published
  - List of currently-running relays
- We use ~100 GiB of archived data
  - Tells us **network state** on any given date since 2005



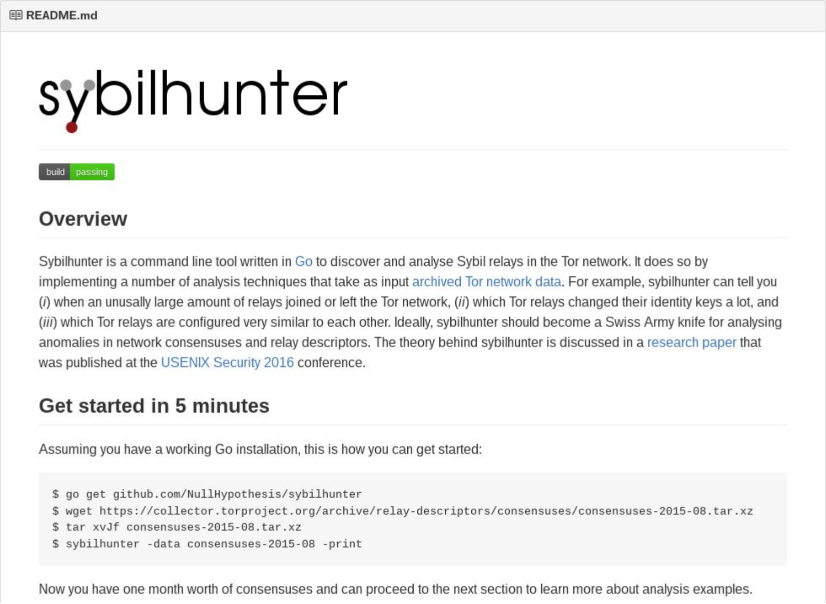
Active dataset

- Used exit relay scanner **exitmap**
  - Runs arbitrary network task over all ~1,000 exit relays
  - Sends decoy traffic over exit relays
- Wrote exitmap modules to detect HTML and HTTP **tampering**
  - Checks if decoy traffic is modified by exit relay
  - Ran modules for 18 months
- Found **251 malicious relays** that serve as ground truth
  - Most of them were **Sybils**
  - Many attempted to **steal Bitcoins**
  - Some injected **JavaScript**



Introducing sybilhunter

- New tool we developed and maintain
  - Freely available at nymity.ch/sybilhunting/
  - ~5,000 lines of code in golang
- Implements **four analysis methods**
  - Network churn
  - Relay uptime visualisation
  - Nearest-neighbour ranking
  - Fingerprint frequency



README.md

sybilhunter

build passing

Overview

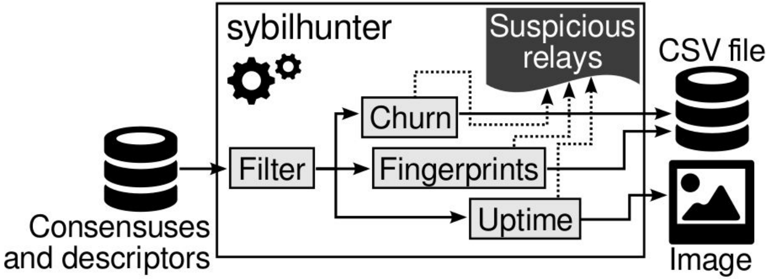
Sybilhunter is a command line tool written in Go to discover and analyse Sybil relays in the Tor network. It does so by implementing a number of analysis techniques that take as input archived Tor network data. For example, sybilhunter can tell you (*i*) when an unusally large amount of relays joined or left the Tor network, (*ii*) which Tor relays changed their identity keys a lot, and (*iii*) which Tor relays are configured very similar to each other. Ideally, sybilhunter should become a Swiss Army knife for analysing anomalies in network consensuses and relay descriptors. The theory behind sybilhunter is discussed in a research paper that was published at the USENIX Security 2016 conference.

Get started in 5 minutes

Assuming you have a working Go installation, this is how you can get started:

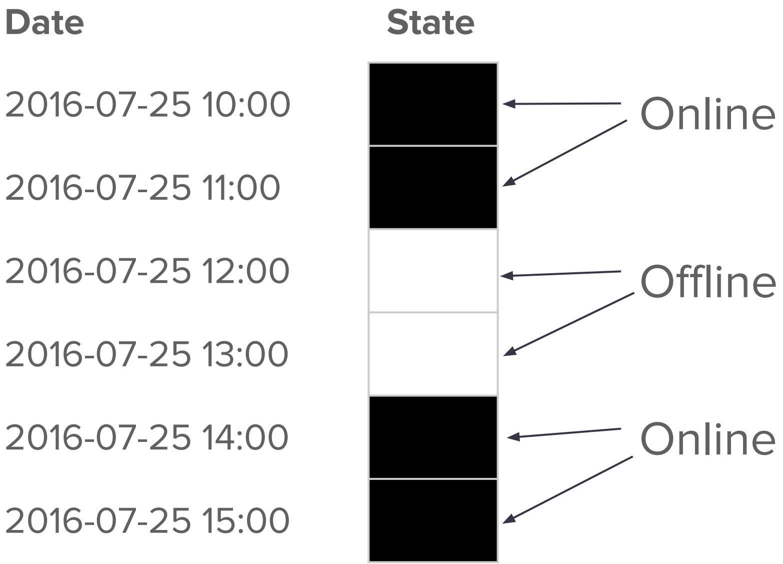
```
$ go get github.com/NullHypothesis/sybilhunter
$ wget https://collector.torproject.org/archive/relay-descriptors/consensuses/consensuses-2015-08.tar.xz
$ tar xvJf consensuses-2015-08.tar.xz
$ sybilhunter -data consensuses-2015-08 -print
```

Now you have one month worth of consensuses and can proceed to the next section to learn more about analysis examples.



Visualizing uptimes (method #1)

- **Each hour**, Tor publishes new consensus
- Allows us to create **binary uptime sequences** for Tor relays



Date	State
2016-07-25 10:00	■
2016-07-25 11:00	■
2016-07-25 12:00	□
2016-07-25 13:00	□
2016-07-25 14:00	■
2016-07-25 15:00	■

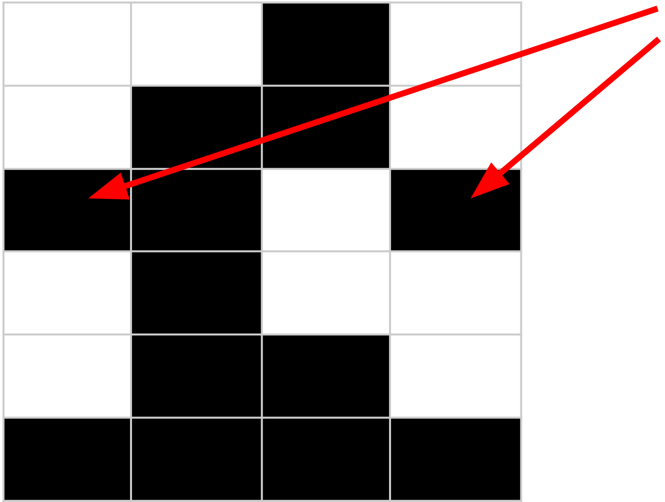
Visualizing uptimes (method #1)

- **Each hour**, Tor publishes new consensus
- Allows us to create **binary uptime sequences** for Tor relays

Date	R_1	R_2	R_3	R_4
2016-07-25 10:00	□	□	■	□
2016-07-25 11:00	□	■	■	□
2016-07-25 12:00	■	■	□	■
2016-07-25 13:00	□	■	□	□
2016-07-25 14:00	□	■	■	□
2016-07-25 15:00	■	■	■	■

Visualizing uptimes (method #1)

- **Each hour**, Tor publishes new consensus
- Allows us to create **binary uptime sequences** for Tor relays

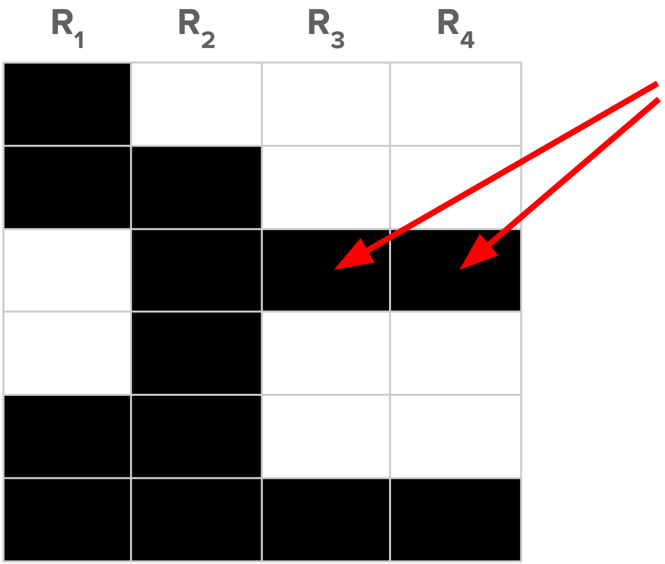


Date	R_1	R_2	R_3	R_4
2016-07-25 10:00	□	□	■	□
2016-07-25 11:00	□	■	■	□
2016-07-25 12:00	■	■	□	■
2016-07-25 13:00	□	■	□	□
2016-07-25 14:00	□	■	■	□
2016-07-25 15:00	■	■	■	■

Critical part is **sorting** columns. We use single-linkage clustering.

Visualizing uptimes (method #1)

- **Each hour**, Tor publishes new consensus
- Allows us to create **binary uptime sequences** for Tor relays

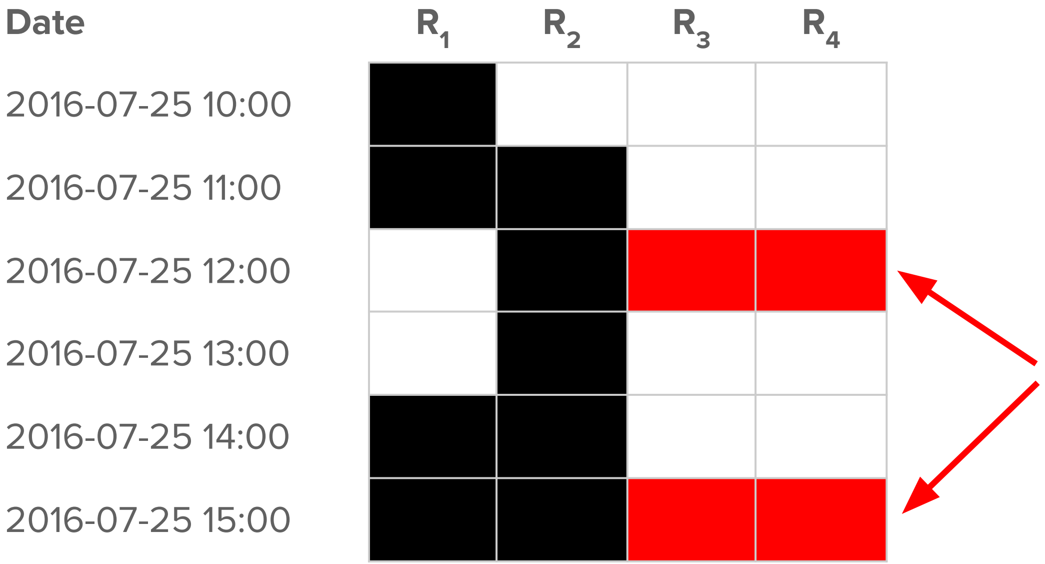


Date	R_1	R_2	R_3	R_4
2016-07-25 10:00	■	□	□	□
2016-07-25 11:00	■	■	□	□
2016-07-25 12:00	□	■	■	■
2016-07-25 13:00	□	■	□	□
2016-07-25 14:00	■	■	□	□
2016-07-25 15:00	■	■	■	■

Sorted columns make it easier to spot Sybils.

Visualizing uptimes (method #1)

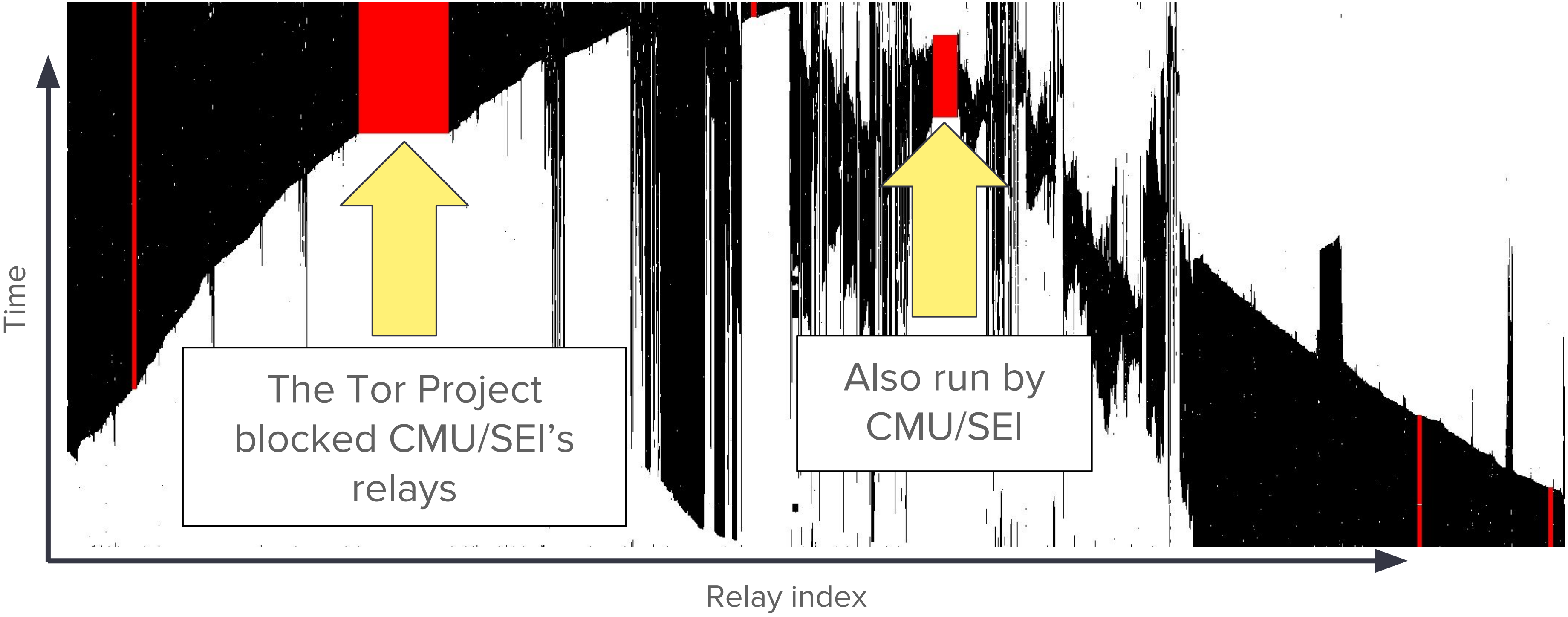
- **Each hour**, Tor publishes new consensus
- Allows us to create **binary uptime sequences** for Tor relays



Date	R_1	R_2	R_3	R_4
2016-07-25 10:00	■			
2016-07-25 11:00	■	■		
2016-07-25 12:00		■	🟥	🟥
2016-07-25 13:00		■		
2016-07-25 14:00	■	■		
2016-07-25 15:00	■	■	🟥	🟥

Highlight identical uptime sequences to facilitate visual inspection

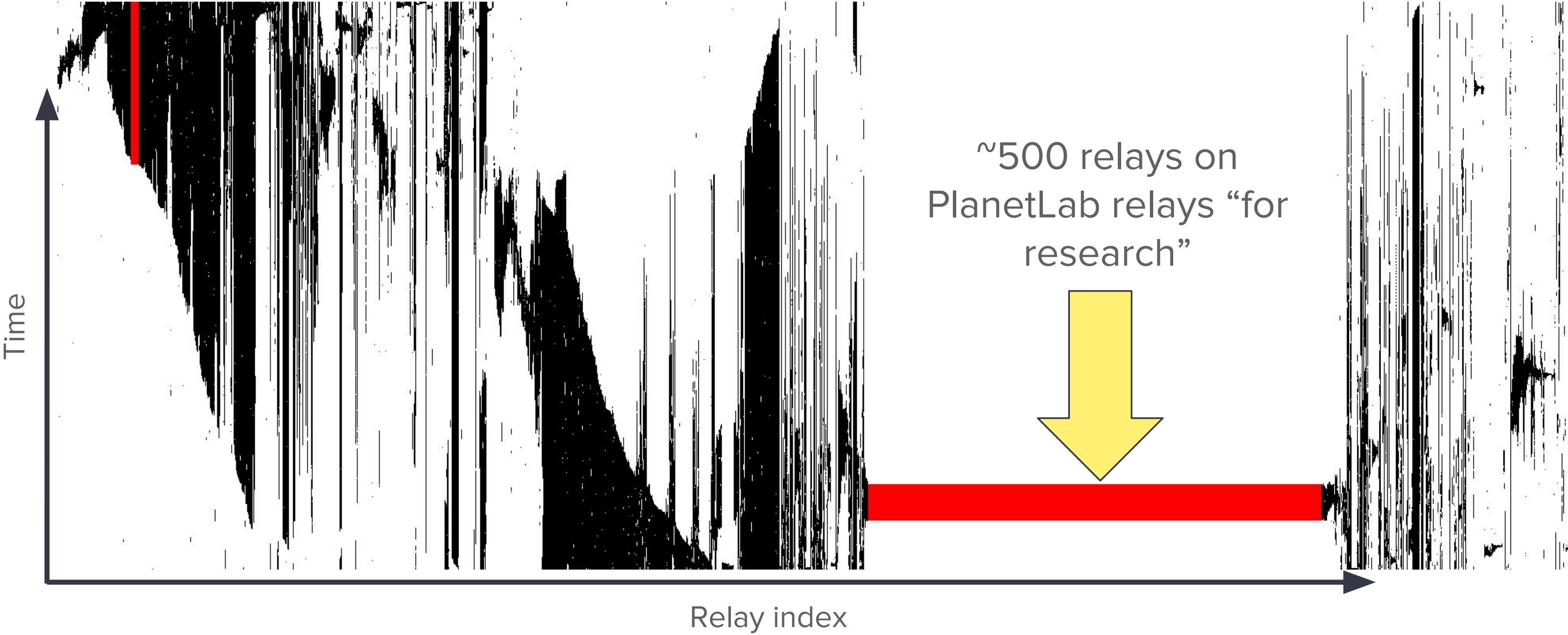
2,034 relays in July 2014



The Tor Project blocked CMU/SEI's relays

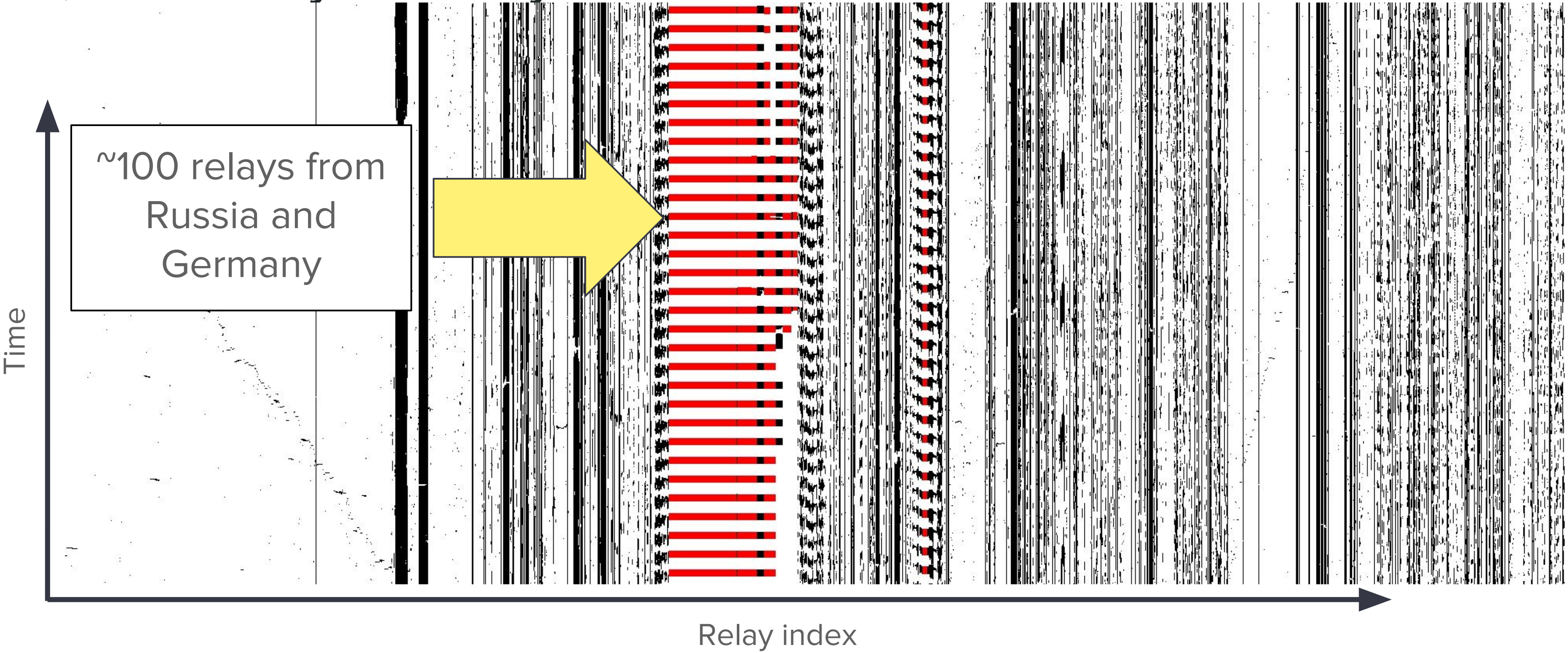
Also run by CMU/SEI

1,629 relays in June 2010



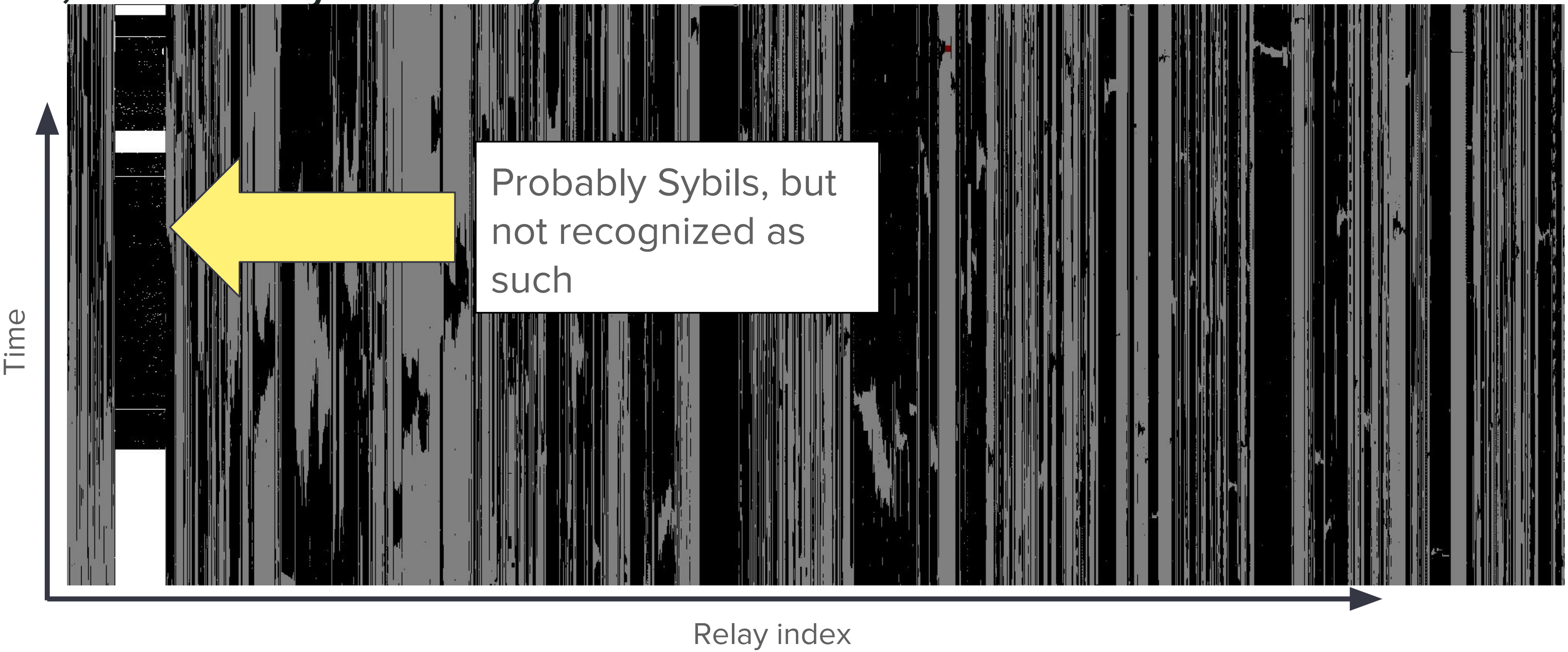
~500 relays on PlanetLab relays “for research”

1,920 relays in July 2012



~100 relays from Russia and Germany

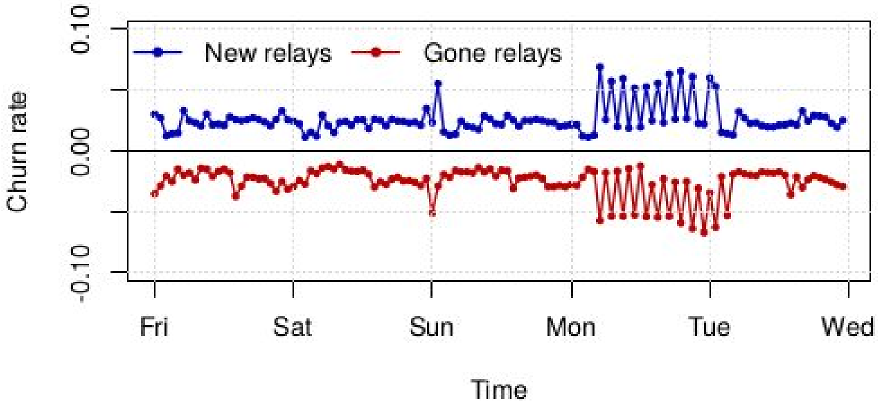
1,920 relays in July 2015



Probably Sybils, but not recognized as such

Network churn (method #2)

- Uptime images provide very **fine-grained** view
- Churn between two subsequent consensuses
  - Each hour, we calculate new churn values
  - $$\text{New churn} = \frac{|\text{Consensus}_t \setminus \text{Consensus}_{t-1}|}{|\text{Consensus}_t|}$$
  - $$\text{Gone churn} = \frac{|\text{Consensus}_{t-1} \setminus \text{Consensus}_t|}{|\text{Consensus}_{t-1}|}$$
- Tor network grew **more stable**
  - Median decreased from 0.04 (2008) to 0.02 (2015)



Changing fingerprints (method #3)

- Generally, Tor relays **don’t change** their fingerprints
  - Fingerprint is 40-digit, relay-specific hash over public key
- Systematic changes can be sign of **DHT manipulation**
- Excerpt from March 2013:
  - `54.242.125.205 (24 unique fingerprints)`
    `54.242.232.162 (24 unique fingerprints)`
    `54.242.42.137  (24 unique fingerprints)`
    `54.242.79.68   (24 unique fingerprints)`
    `54.242.248.129 (24 unique fingerprints)`
    `54.242.151.229 (24 unique fingerprints)`
    `54.242.198.54  (24 unique fingerprints)`
  - See S&P’13 paper “Trawling for Tor Hidden Services”

2013 IEEE Symposium on Security and Privacy

Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization

Alex Biryukov, Ivan Pustogarov, Ralf-Philipp Weinmann
University of Luxembourg
{alex.biryukov,ivan.pustogarov,ralf-philipp.weinmann}@uni.lu

***Abstract*—Tor is the most popular volunteer-based anonymity network consisting of over 3000 volunteer-operated relays. Apart from making connections to servers hard to trace to their origin it can also provide receiver privacy for Internet services through a feature called “hidden services”.**

In this paper we expose flaws both in the design and implementation of Tor’s hidden services that allow an attacker to measure the popularity of arbitrary hidden services, take down hidden services and deanonymize hidden services. We give a practical evaluation of our techniques by studying: (1) a recent case of a botnet using Tor hidden services for command and control channels; (2) Silk Road, a hidden service used to sell drugs and other contraband; (3) the hidden service of the DuckDuckGo search engine.

***Keywords*-Tor; anonymity network; privacy; hidden services**

I. Introduction

Research into low-latency anonymity networks has mostly focused on sender anonymity, i.e. allowing users to connect to network resources without disclosing their network address to the destination. Nonetheless, to guarantee freedom of speech, responder privacy[1] is equally if not even more important. Allowing the people not only to access information anonymously but also to publish anonymously is an important aspect of nurturing democracy. Similarly, providing internet services without disclosing their location and owner puts a constraint on what attacks can be performed by adversaries.

Tor [7] arguably is the most popular and well-researched low-latency anonymity network, providing sender privacy for internet services to its users; additionally responder privacy can be achieved with Tor by making TCP services available as *hidden services*. While the first generation of Tor’s hidden service design has been described in the original design paper, the current version of Tor is using a revised design [25]. Justifications of the design choices and attack scenarios considered are given in [15].

This paper analyzes the security of Tor hidden services. We look at them from different attack perspectives and provide a systematic picture of what information can be obtained with very inexpensive means. We focus both on attacks that allow to censor access to targeted hidden services as well as on deanonymization of hidden services. As the result we believe that many components of the current Tor HS protocol should be improved and while a short term patch may mitigate some of the problems, a more complex approach is required both in terms of efficiency and in terms of privacy.

We also study *deployed hidden services*. For instance, we apply our findings to a botnet which makes its command and control center available to bots as a Tor hidden service (we extracted its onion address by analyzing a malware sample) and extrapolate its size by counting the number of hidden service requests.

Contributions:

- We give a method to measure the popularity of any hidden service without the consent of the hidden service operator.
- We show how connectivity to selected hidden services can be denied by impersonating all of their responsible hidden services directories.
- We demonstrate a technique that allows one to harvest hidden service descriptors (and thus get a global picture of all hidden services in Tor) in approximately 2 days using only a modest amount of resources.
- We show how to reveal the guard nodes of a Tor hidden service.
- We propose a large-scale opportunistic deanonymization attack, capable of revealing IP addresses of a significant fraction of Tor’s hidden services over a one year period of time.

Most of our attacks are made practical and cost efficient by two implementation deficiencies in the current versions of Tor[2]: (1) Tor relays can cheat and inflate their bandwidth in the consensus despite bandwidth measurements; this makes them more likely to be chosen by the path selection algorithm. (2) Using a technique called “shadowing” we can phase relays in and out of the consensus at will without them losing their flags, allowing us to defeat countermeasures against Sybil [8] attacks.

Ethical considerations: Attacks against Tor can be simulated in dedicated simulators such as Shadow [13]. However, deployed hidden services are not well studied. Until now there have been no statistics about the number of hidden services or their usage statistics. Henceforth, we deem experiments on the live Tor network that do

[1] Traditionally called recipient privacy in the case of mix networks.

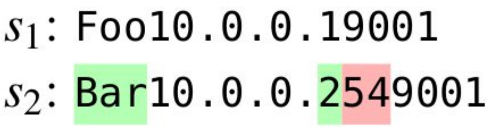
[2] We consider versions up to tor v0.2.4.6-alpha, which was the most current version at the time of submission.

1081-6011/13 $26.00 © 2013 IEEE
DOI 10.1109/SP.2013.15

IEEE computer society

Nearest neighbour ranking (method #4)

- Exitmap occasionally discovered **malicious relays**
  - Were there **more**, but we failed to find them?
  - Given relay R_1, what are its most similar “neighbours”?
- Rank relay’s nearest neighbour by configuration similarity
  - First, turn relay configurations into string
  - Then, calculate Levenshtein distance to “reference” relay
- Example of Levenshtein distance being six
  - Four modifications
  - Two deletions

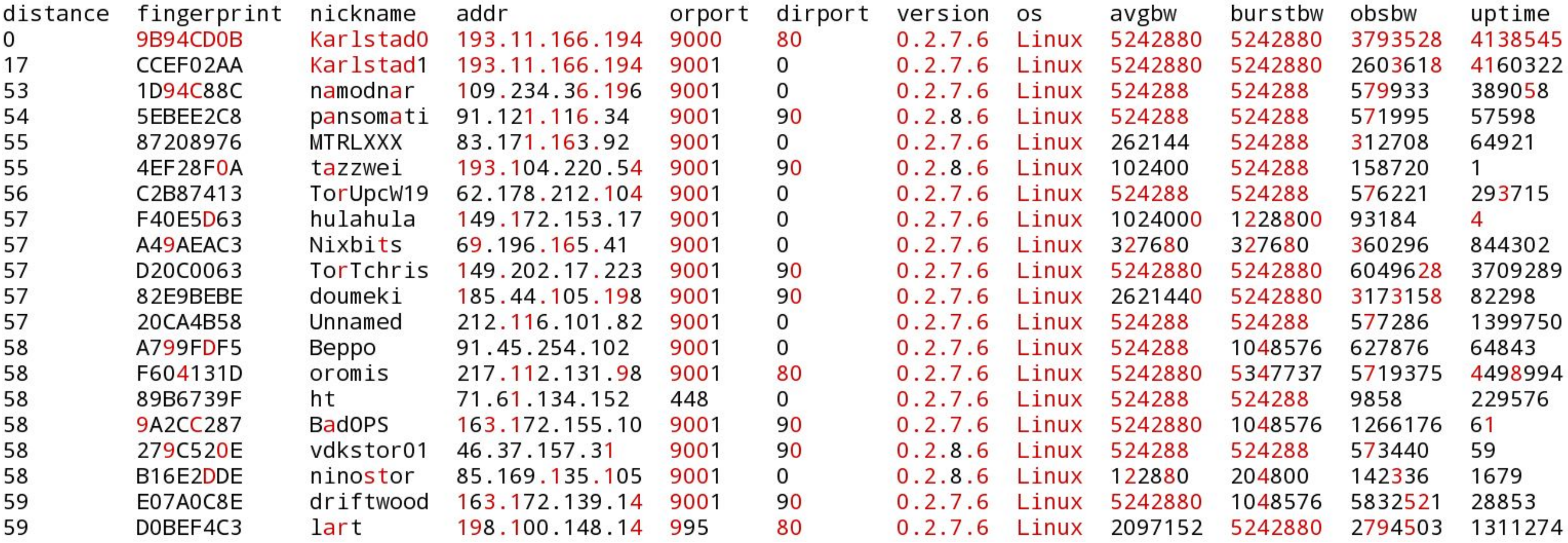


s_1: Foo10.0.0.19001

s_2: Bar10.0.0.2549001

Nearest neighbour search in action

- Tool available at nymity.ch/sybilhunting/



```
distance  fingerprint  nickname   addr             orport  dirport  version  os     avgbw    burstbw  obsbw    uptime
0         9B94CD0B     Karlstad0  193.11.166.194   9000    80       0.2.7.6  Linux  5242880  5242880  3793528  4138545
17        CCEF02AA     Karlstad1  193.11.166.194   9001    0        0.2.7.6  Linux  5242880  5242880  2603618  4160322
53        1D94C88C     namodnar   109.234.36.196   9001    0        0.2.7.6  Linux  524288   524288   579933   389058
54        5EBEE2C8     pansomati  91.121.116.34    9001    90       0.2.8.6  Linux  524288   524288   571995   57598
55        87208976     MTRLXXX    83.171.163.92    9001    0        0.2.7.6  Linux  262144   524288   312708   64921
55        4EF28F0A     tazzwei    193.104.220.54   9001    90       0.2.8.6  Linux  102400   524288   158720   1
56        C2B87413     TorUpcW19  62.178.212.104   9001    0        0.2.7.6  Linux  524288   524288   576221   293715
57        F40E5D63     hulahula   149.172.153.17   9001    0        0.2.7.6  Linux  1024000  1228800  93184    4
57        A49AEAC3     Nixbits    69.196.165.41    9001    0        0.2.7.6  Linux  327680   327680   360296   844302
57        D20C0063     TorTchris  149.202.17.223   9001    90       0.2.7.6  Linux  5242880  5242880  6049628  3709289
57        82E9BEBE     doumeki    185.44.105.198   9001    90       0.2.7.6  Linux  2621440  5242880  3173158  82298
57        20CA4B58     Unnamed    212.116.101.82   9001    0        0.2.7.6  Linux  524288   524288   577286   1399750
58        A799FDF5     Beppo      91.45.254.102    9001    0        0.2.7.6  Linux  524288   1048576  627876   64843
58        F604131D     oromis     217.112.131.98   9001    80       0.2.7.6  Linux  5242880  5347737  5719375  4498994
58        89B6739F     ht         71.61.134.152    448     0        0.2.7.6  Linux  524288   524288   9858     229576
58        9A2CC287     BadOPS     163.172.155.10   9001    90       0.2.7.6  Linux  5242880  1048576  1266176  61
58        279C520E     vdkstor01  46.37.157.31     9001    90       0.2.8.6  Linux  524288   524288   573440   59
58        B16E2DDE     ninostor   85.169.135.105   9001    0        0.2.8.6  Linux  122880   204800   142336   1679
59        E07A0C8E     driftwood  163.172.139.14   9001    90       0.2.7.6  Linux  5242880  1048576  5832521  28853
59        D0BEF4C3     lart       198.100.148.14   995     80       0.2.7.6  Linux  2097152  5242880  2794503  1311274
```

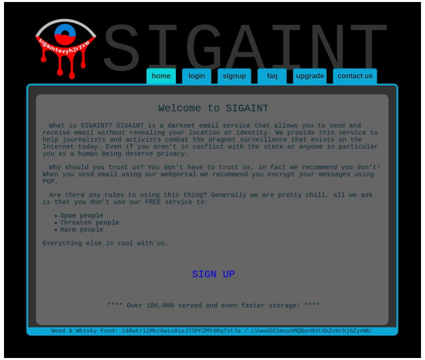
Our results in a nutshell

- Studied **twenty** Sybil groups ➔ lower bound

Purpose	# of Sybil groups	Description
MitM	7	Attempted to steal Bitcoins by manipulating Tor exit traffic
Botnet	2	Relays seemed part of botnet
DoS	1	Attempted to (unsuccessfully) disable Tor network
Research	4	Various live experiments, mostly on hidden services
Unknown	6	Purpose unclear, perhaps benign

Discussion of “Bitcoin Sybils”

- Attempted to **steal Bitcoins** from Tor users
  - All Sybils were exit relays
  - Transparent rewriting of Bitcoin addresses
- **Resurfaced** after The Tor Project blocked relays
  - Game of whack-a-mole
  - Went on for many **months**



Welcome to SIGAINT

What is SIGAINT? SIGAINT is a darknet email service that allows you to send and receive email without revealing your location or identity. We provide this service to help journalists and activists combat the dragnet surveillance that exists on the Internet today. Even if you aren't in conflict with the state or anyone in particular you as a human being deserve privacy.

Why should you trust us? You don't have to trust us, in fact we recommend you don't! When you send email using our webportal we recommend you encrypt your messages using PGP.

Are there any rules to using this thing? Generally we are pretty chill, all we ask is that you don't use our FREE service to:

- Spam people
- Threaten people
- Harm people

Everything else is cool with us.

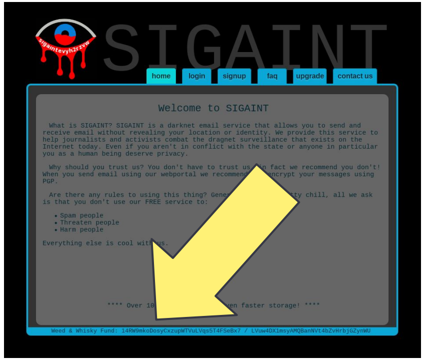
SIGN UP

**** Over 100,000 served and even faster storage! ****

Weed & Whisky Fund: 14Rwtr11Mkc6wix9isJ7SPFZMY4Rq7st7a / LVuw4DX1msyAMQBanNVt4bZvHrbjGZynWU

Original:

14Rwtr11Mkc6wix9isJ7SPFZMY4Rq7st7a



Welcome to SIGAINT

What is SIGAINT? SIGAINT is a darknet email service that allows you to send and receive email without revealing your location or identity. We provide this service to help journalists and activists combat the dragnet surveillance that exists on the Internet today. Even if you aren't in conflict with the state or anyone in particular you as a human being deserve privacy.

Why should you trust us? You don't have to trust us in fact we recommend you don't! When you send email using our webportal we recommend encrypt your messages using PGP.

Are there any rules to using this thing? Gene tty chill, all we ask is that you don't use our FREE service to:

- Spam people
- Threaten people
- Harm people

Everything else is cool wit us.

**** Over 10 en faster storage! ****

Weed & Whisky Fund: 14RW9mkoDosyCxzupWTVuLVqs5T4FSeBx7 / LVuw4DX1msyAMQBanNVt4bZvHrbjGZynWU

Fake:

14RW9mkoDosyCxzupWTVuLVqs5T4FSeBx7

Limitations

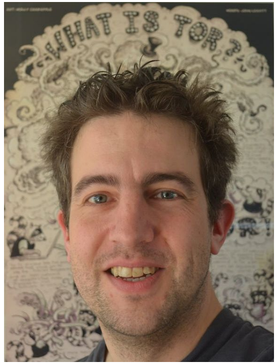
- **Determining intent** is hard
- Our results are a **lower bound**
- Sybilhunter works best against **ignorant** attacker
    - **Open** analysis framework, **secret** parameters
- Hard to exposure **future** attacks

Discussion

- Our adversaries are often **lazy** and we can **exploit** that
- **Different types** of Sybils call for **different methods**
- Academic research not **harmless by definition**
  - research.torproject.org/safetyboard.html
- Methods are **general** and apply to **other networks** as well
- Crowdsourcing successful

Acknowledgements

Roya Ensafi



Karsten Loesing

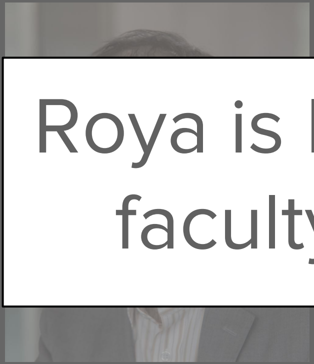
Nick Feamster

- Thanks to
  - Georg Koppen
  - Prateek Mittal
  - Stefan Lindskog
  - Tor developers and community
  - Tudor Dumitraş (our shepherd)
- Open code, data, visualisations:
  - nymity.ch/sybilhunting/
- Contact
  - phw@nymity.ch
  - @__phw

Acknowledgements

Roya Ensafi

Karsten Loesing



Nick Feamster

Roya is looking for a faculty position!

- Thanks to
  - Tor developers and community
  - Tudor Dumitraş (our shepherd)
- Open code, data, visualisations:
  - `nymity.ch/sybilhunting/`
- Contact
  - `phw@nymity.ch`
  - `@__phw`