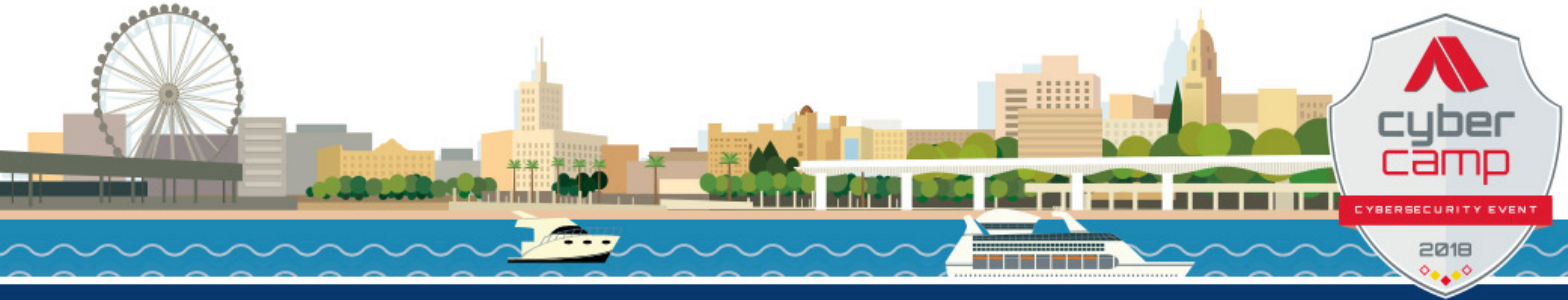
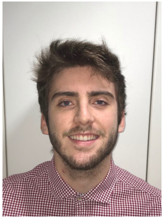
On Tor protocol modeling and characterization of Hidden Services

Ricardo J. Rodríguez (CUD), Jorge García de Quirós (UZ)



$whoami

- **CLS member** (2001)
- **Ph.D. in Compt. Sci.** (2013)
- **Assistant Professor at Centro Universitario de la Defensa**, AGM (Zaragoza)
- Research lines
  - Security-driven engineering
  - Survivability analysis
  - Program binary analysis
  - RFID/NFC security
- Email: rjrodriguez@unizar.es



- **BSc. in Compt. Sci.** (2018)
- **Junior researcher at University of Zaragoza**, Spain
- Email: jgarciaqg@unizar.es

Agenda

1. Introduction
2. Modeling the Tor Network
   - Elements
   - Connections
   - Node Behaviors
3. Towards the Deanonymization of Tor Hidden Services
   - Description of `TorHSScanner`
   - Experimental Results
4. Related Work
5. Conclusions

Agenda

1 Introduction

2 Modeling the Tor Network

3 Towards the Deanonymization of Tor Hidden Services

4 Related Work

5 Conclusions

Introduction

- **Network specially designed to improve user’s anonymity and privacy on the Internet**
- **Developed by the US Naval Research Laboratory, aiming at protecting government communications.** Today:
  - Communicate whist-blowers with journalists anonymously
  - Activists in conflict zones
  - Undercover or surveillance operations
  - Other strategic or military purposes

Introduction

- Currently maintained by an ONG (*The Tor Project*). They do:
  - Manage the development of the Tor browser
  - Manage the development of the Tor protocols
  - Control the network status

- **Network specially designed to improve user's anonymity and privacy on the Internet**
- **Developed by the US Naval Research Laboratory, aiming at protecting government communications.** Today:
  - Communicate whist-blowers with journalists anonymously
  - Activists in conflict zones
  - Undercover or surveillance operations
  - Other strategic or military purposes

Introduction

How does it work?

- **Anonymous low-latency communication based on virtual circuits**
  - **Intermediate hops**: every hop is a node
  - **Virtual circuits**: they guarantee there DOES NOT exist a direct connection between the server and the client. Every node only knows about the next and the previous
  - **Network traffic is ciphered in layers → onion routing**

Introduction

How does it work?

- **Anonymous low-latency communication based on virtual circuits**
  - **Intermediate hops**: every hop is a node
  - **Virtual circuits**: they guarantee there DOES NOT exist a direct connection between the server and the client. Every node only knows about the next and the previous
  - **Network traffic is ciphered in layers → onion routing**

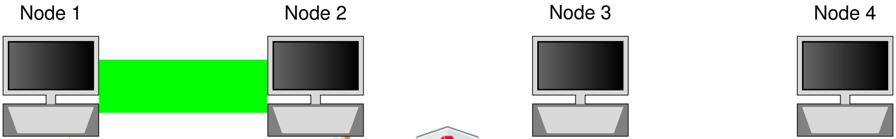
Suppose Node 1 wants to communicate with Node 4

Introduction

How does it work?

- **Anonymous low-latency communication based on virtual circuits**
  - **Intermediate hops**: every hop is a node
  - **Virtual circuits**: they guarantee there DOES NOT exist a direct connection between the server and the client. Every node only knows about the next and the previous
  - **Network traffic is ciphered in layers** → **onion routing**

Suppose Node 1 wants to communicate with Node 4

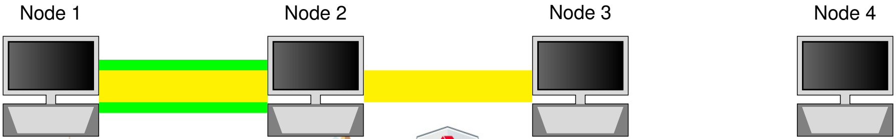


Introduction

How does it work?

- **Anonymous low-latency communication based on virtual circuits**
  - **Intermediate hops**: every hop is a node
  - **Virtual circuits**: they guarantee there DOES NOT exist a direct connection between the server and the client. Every node only knows about the next and the previous
  - **Network traffic is ciphered in layers → onion routing**

Suppose Node 1 wants to communicate with Node 4

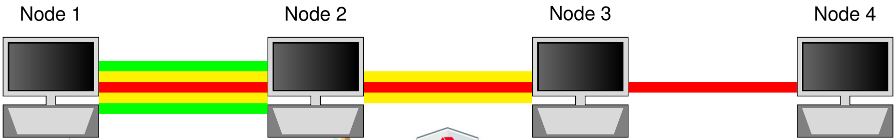


Introduction

How does it work?

- **Anonymous low-latency communication based on virtual circuits**
  - **Intermediate hops**: every hop is a node
  - **Virtual circuits**: they guarantee there DOES NOT exist a direct connection between the server and the client. Every node only knows about the next and the previous
  - **Network traffic is ciphered in layers → onion routing**

Suppose Node 1 wants to communicate with Node 4



Introduction

Why do we need Tor?

Introduction

Why do we need Tor?

Introduction

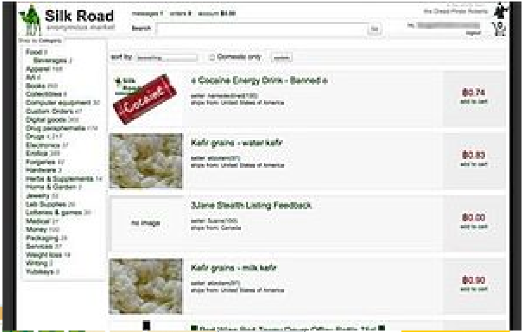
Why do we need Tor?

- **Legitimate needs: defense of individual rights and freedoms**
  - Access to censured stuff in the country where a user lives in (e.g., Tienanmen Square massacre)
  - Access to chat rooms or forums to victims of sexual abuses and/or rapes
  - Anonymous public protests of human/civil right violations of states/companies

Introduction

Why do we need Tor?

- **Illegitimate uses: cybercriminals also use Tor for their businesses**
  - Well-known example: *SilkRoad* (online drug market). Launched in 2011, closed in 2013 by the FBI. **His founder/owner is currently serving a double life sentence plus forty years, without the possibility of parole**



Silk Road
anonymous market

Search

Shop by Category
Food
Beverages
Apparel
Art
Books
Collectibles
Computer equipment
Custom Orders
Digital goods
Drug paraphernalia
Drugs
Electronics
Erotica
Forgeries
Hardware
Herbs & Supplements
Home & Garden
Jewelry
Lab Supplies
Lotteries & games
Medical
Money
Packaging
Services
Weight loss
Writing
Yubikeys

sort by

○ Domestic only

♦ Cocaine Energy Drink - Banned ♦
ships from United States of America
฿0.74

Kefir grains - water kefir
ships from United States of America
฿0.83

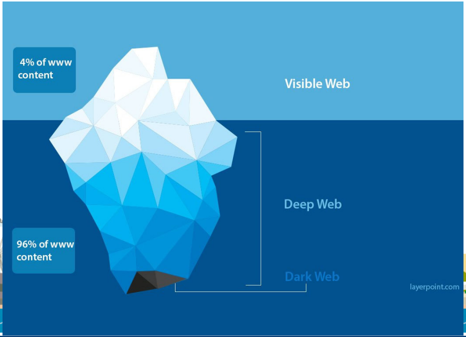
3Jane Stealth Listing Feedback
no image
ships from Canada
฿0.00

Kefir grains - milk kefir
ships from United States of America
฿0.90

Introduction – surface vs. deep vs. dark web

Hidden services

- **Services only provided through the Tor network**
- **Double blind.** We do not know where the server is located and who accesses to it
- `.onion address`. Recognized by IETF/IANA in 2015



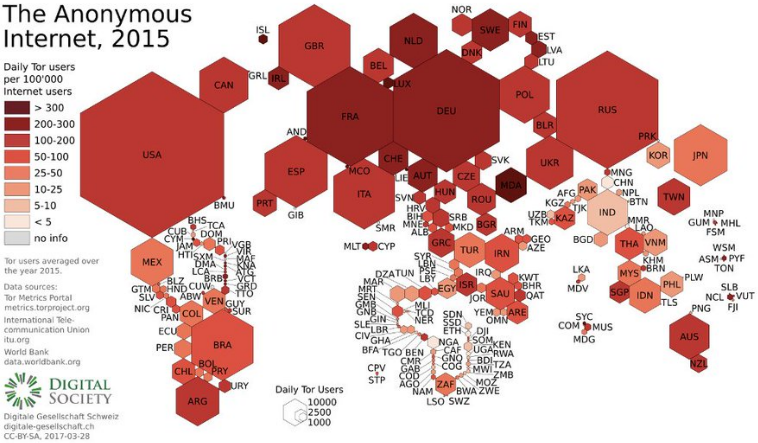
Some statistics (19/11/18):

`https://metrics.torproject.org/`

- ≈6.5K nodes
- ≈2M of unique users
- ≈115K hidden services

Introduction – Map of daily users during 2015

Credits: https://mobile.twitter.com/torproject/status/877556893941628928



The Anonymous Internet, 2015

Daily Tor users per 100'000 Internet users

- > 300
- 200-300
- 100-200
- 50-100
- 25-50
- 10-25
- 5-10
- < 5
- no info

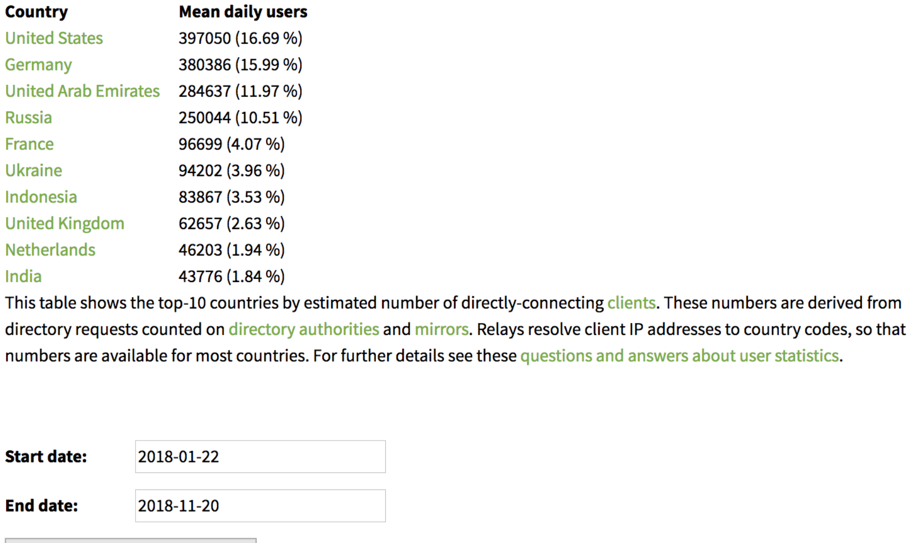
Tor users averaged over the year 2015.

Data sources:
Tor Metrics Portal
metrics.torproject.org
International Telecommunication Union
itu.org
World Bank
data.worldbank.org

DIGITAL SOCIETY
Digitale Gesellschaft Schweiz
digitale-gesellschaft.ch
CC-BY-SA, 2017-03-28

Daily Tor Users
10000
2500
1000

Introduction – Top-10 relay users (per country)



Country	Mean daily users
United States	397050 (16.69 %)
Germany	380386 (15.99 %)
United Arab Emirates	284637 (11.97 %)
Russia	250044 (10.51 %)
France	96699 (4.07 %)
Ukraine	94202 (3.96 %)
Indonesia	83867 (3.53 %)
United Kingdom	62657 (2.63 %)
Netherlands	46203 (1.94 %)
India	43776 (1.84 %)

This table shows the top-10 countries by estimated number of directly-connecting clients. These numbers are derived from directory requests counted on directory authorities and mirrors. Relays resolve client IP addresses to country codes, so that numbers are available for most countries. For further details see these questions and answers about user statistics.

Start date: 2018-01-22

End date: 2018-11-20

Agenda

1 Introduction

2 Modeling the Tor Network

- Elements
- Connections
- Node Behaviors

3 Towards the Deanonymization of Tor Hidden Services

4 Related Work

5 Conclusions

Modeling the Tor Network

- Lot of works working in **privacy and anonymity aspects**
  - Check the incredible stuff at `https://www.freehaven.net/anonbib/`
- However, **none studies the modeling and formalization of the Tor protocol**
  - Note here: by Tor protocol we mean *every protocol used in the Tor network*
  - Public documentation (`https://gitweb.torproject.org/torspec.git/tree`)
  - Divided into 20 text files (*old scholz*, no format, no graphics, an average of ≈ 18000 text lines per file

Modeling the Tor Network

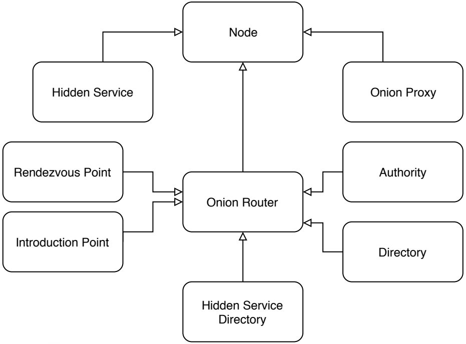
- Lot of works working in **privacy and anonymity aspects**
  - Check the incredible stuff at `https://www.freehaven.net/anonbib/`
- However, **none studies the modeling and formalization of the Tor protocol**
  - Note here: by Tor protocol we mean *every protocol used in the Tor network*
  - Public documentation (`https://gitweb.torproject.org/torspec.git/tree`)
  - Divided into 20 text files (*old scholz*, no format, no graphics, an average of ≈ 18000 text lines per file

Contribution

- **Modeling of the Tor protocol with UML**
  - **First goal to find, if exists, any design flaw using model verification**
  - **Success in other domains** (e.g., EMV cards, Modbus protocol)

Elements of the Tor network

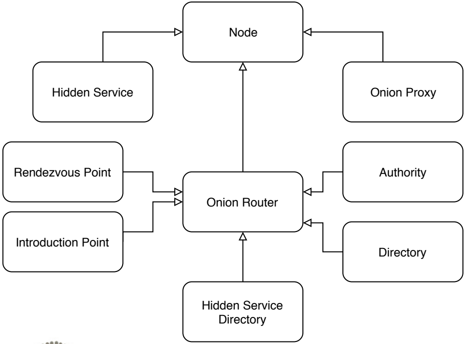
Types of nodes



- **Onion Proxy** (OP): **Tor client**
- **Onion Router** (OR)
  - **Basic element of the Tor network**
  - They use a default port (9050)
  - Maintained by volunteers, they establish the virtual circuits to connect
  - Depending on the position in the circuit, we distinguish:
    - **Guard node**: first node of the circuit
    - **Middle node**: any position in the circuit (but first and last)
    - **Exit node**: last node of the circuit
- **Hidden Service** (HS)
  - **Services only available through the Tor network**
  - TLD address `.onion` (approved by IETF/IANA in 2015)

Elements of the Tor network

Types of nodes



Node

Hidden Service

Onion Proxy

Rendezvous Point

Onion Router

Authority

Introduction Point

Directory

Hidden Service Directory

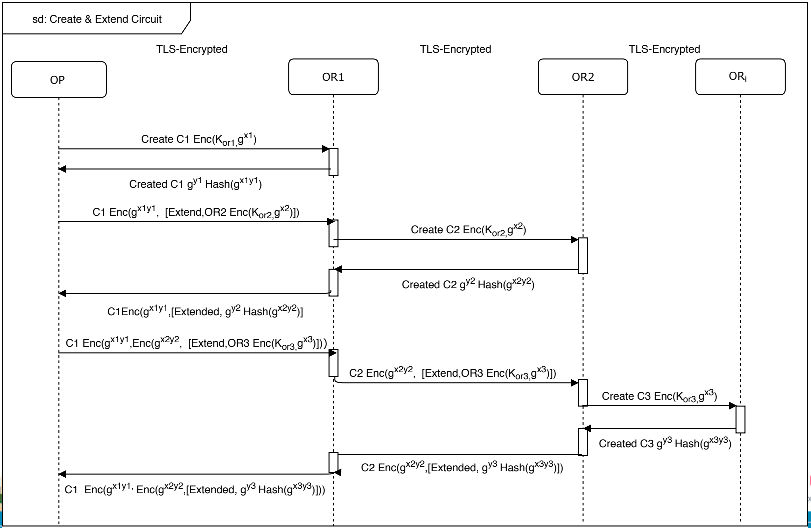
- **Directory** (Dir): they obtain information current status of the network
- **Authority** (Auth): Dir with authority permission
  - Minority role. It needs a long uptime and good performance
  - Chose by the organization. Only 10
  - Elaborate the current status of the network
- **Introduction Point** (IP) and **Rendezvouz** (RV)
  - Specific tasks when connecting to a HS (next slides!)
- **Bridge**
  - **Hidden OR node** (not listed)
  - First node in countries where the use of Tor is prohibited
- **HS directory** (HsDir)
  - They store how to connect to the HS

Elements of the Tor network

Types of messages

- **Control Cell**
  - **Header + payload**
  - Contains **information about the circuit identification and the command** (action to perform)
  - Interpreted always directly by the receiver of the cells
  - Examples: CREATE, CREATED, DESTROY o PADDING
- **Relay Cell**
  - **Header + additional header + payload**
  - **The additional header helps to identify the data flow of that cell**
    - A data flow goes through a Tor circuit. A Tor circuit may contain different data flows
  - Examples: DATA, BEGIN, END, CONNECTED, EXTEND, o EXTENDED

Connections – Virtual circuit establishment



sd: Create & Extend Circuit

TLS-Encrypted

TLS-Encrypted

TLS-Encrypted

OP

OR1

OR2

OR_i

Create C1 Enc(K_{or1},g^{x1})

Created C1 g^{y1} Hash(g^{x1y1})

C1 Enc(g^{x1y1}, [Extend,OR2 Enc(K_{or2},g^{x2})])

Create C2 Enc(K_{or2},g^{x2})

Created C2 g^{y2} Hash(g^{x2y2})

C1Enc(g^{x1y1},[Extended, g^{y2} Hash(g^{x2y2})]

C1 Enc(g^{x1y1},Enc(g^{x2y2}, [Extend,OR3 Enc(K_{or3},g^{x3})]))

C2 Enc(g^{x2y2}, [Extend,OR3 Enc(K_{or3},g^{x3})])

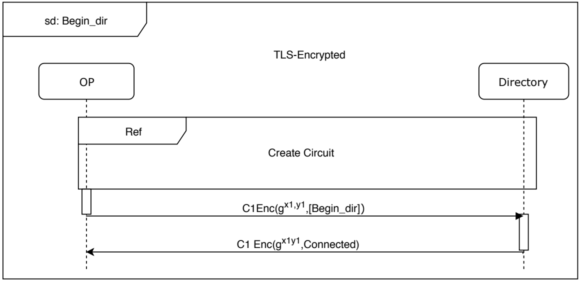
Create C3 Enc(K_{or3},g^{x3})

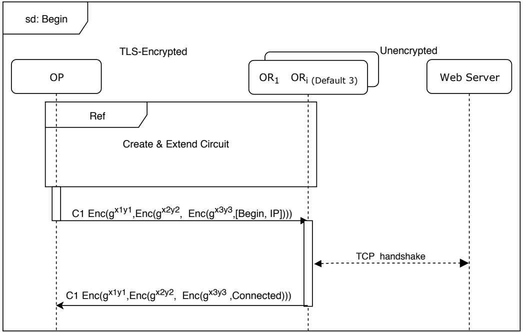
Created C3 g^{y3} Hash(g^{x3y3})

C2 Enc(g^{x2y2},[Extended, g^{y3} Hash(g^{x3y3})])

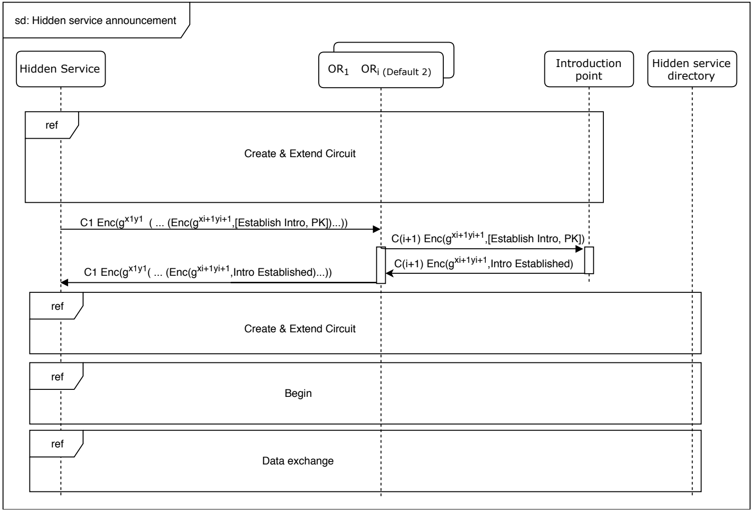
C1 Enc($g^{x1y1,}$ Enc(g^{x2y2},[Extended, g^{y3} Hash(g^{x3y3})]))

Connections – Internal and external communications

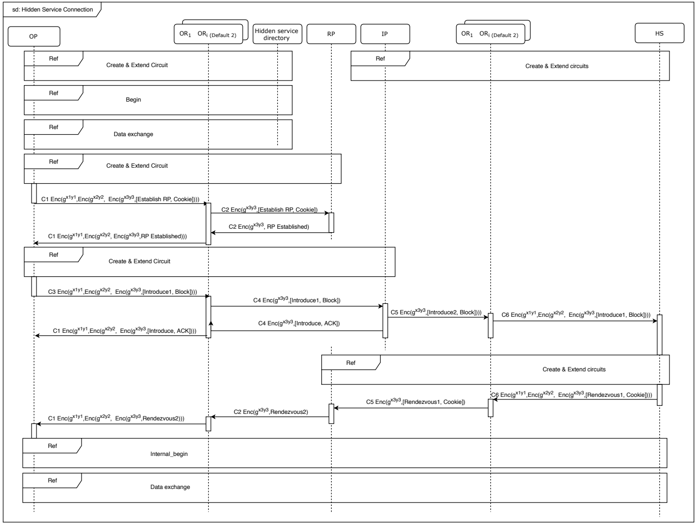



sd: Begin
TLS-Encrypted
Unencrypted
OP
OR_1 OR_i (Default 3)
Web Server
Ref
Create & Extend Circuit
C1 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},[Begin, IP])))
TCP handshake
C1 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3} ,Connected)))

Connection to a HS – HS announce


sd: Hidden service announcement
Hidden Service
OR_1 OR_i (Default 2)
Introduction point
Hidden service directory
ref
Create & Extend Circuit
C1 Enc(g^{x1y1} (... (Enc($g^{xi+1yi+1}$,[Establish Intro, PK])...))
C(i+1) Enc($g^{xi+1yi+1}$,[Establish Intro, PK])
C(i+1) Enc($g^{xi+1yi+1}$,Intro Established)
C1 Enc(g^{x1y1}(... (Enc($g^{xi+1yi+1}$,Intro Established)...))
ref
Create & Extend Circuit
ref
Begin
ref
Data exchange

Connection to a HS – HS connection



sd: Hidden Service Connection

OP

OR_1 OR_i (Default 2)

Hidden service directory

RP

IP

OR_1 OR_i (Default 2)

HS

Ref Create & Extend Circuit

Ref Create & Extend circuits

Ref Begin

Ref Data exchange

Ref Create & Extend Circuit

C1 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},[Establish RP, Cookie])))

C2 Enc(g^{x3y3},[Establish RP, Cookie])

C2 Enc(g^{x3y3}, RP Established)

C1 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},RP Established)))

Ref Create & Extend Circuit

C3 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},[Introduce1, Block])))

C4 Enc(g^{x3y3},[Introduce1, Block])

C5 Enc(g^{x3y3},[Introduce2, Block])))

C6 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},[Introduce1, Block])))

C4 Enc(g^{x3y3},[Introduce, ACK])

C1 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},[Introduce, ACK])))

Ref Create & Extend circuits

C6 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},[Rendezvous1, Cookie])))

C5 Enc(g^{x3y3},[Rendezvous1, Cookie])

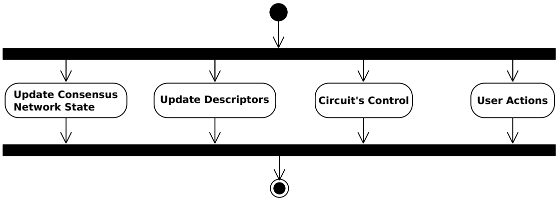
C2 Enc(g^{x3y3},Rendezvous2)

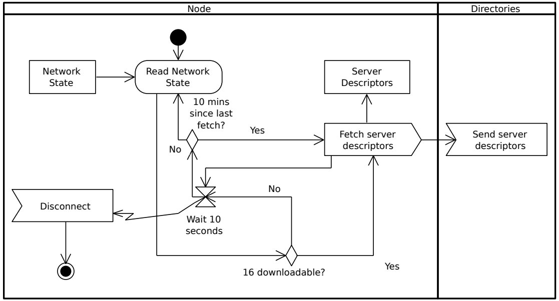
C1 Enc(g^{x1y1},Enc(g^{x2y2}, Enc(g^{x3y3},Rendezvous2)))

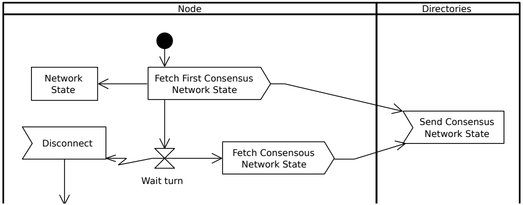
Ref Internal_begin

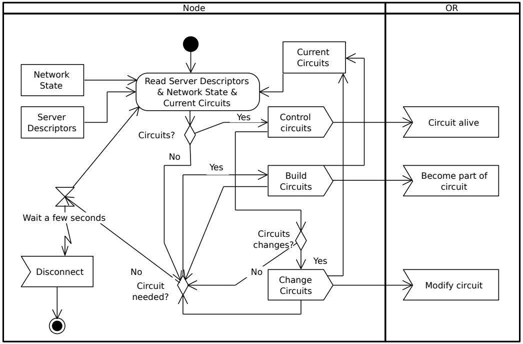
Ref Data exchange

Node Behaviors – default behavior



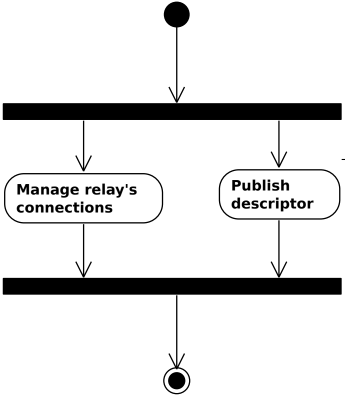



Node
Directories
Network State
Fetch First Consensus Network State
Send Consensus Network State
Disconnect
Fetch Consensous Network State
Wait turn



Node Behaviors – OR, Auth, and Dir nodes

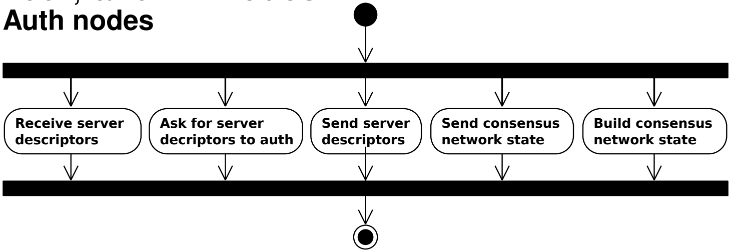
OR nodes



Manage relay's connections

Publish descriptor

Auth nodes



Receive server descriptors

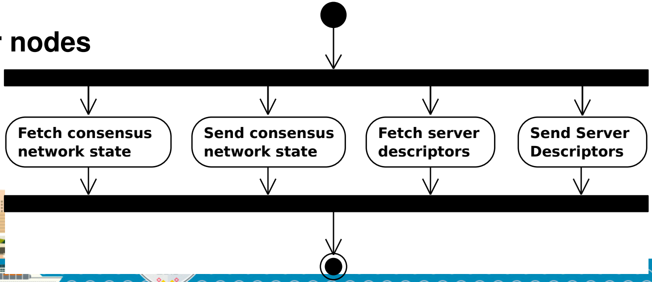
Ask for server decriptors to auth

Send server descriptors

Send consensus network state

Build consensus network state

Dir nodes



Fetch consensus network state

Send consensus network state

Fetch server descriptors

Send Server Descriptors

Agenda

1 Introduction

2 Modeling the Tor Network

3 Towards the Deanonymization of Tor Hidden Services
- Description of `TorHSScanner`
- Experimental Results

4 Related Work

5 Conclusions

Towards the Deanonymization of Tor Hidden Services

Contribution

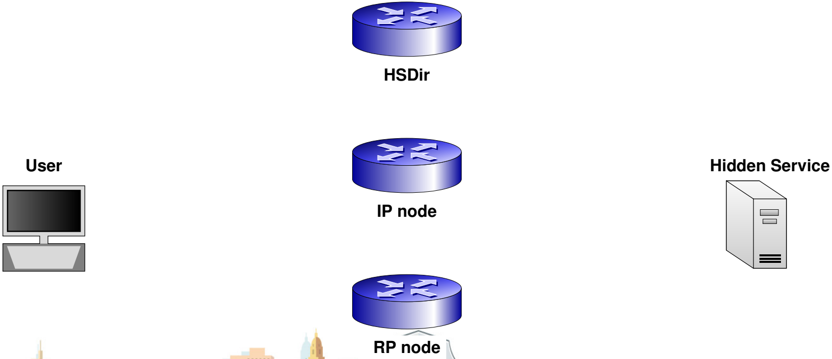
- `TorHSScanner`: **automatic system to access to a Tor hidden service and retrieve some characteristics to deanonymize it**
  - 1796 hidden service addresses collected
  - 346 bounded to an (visible) Internet system with similar characteristics
- **Categorization of Tor hidden services**
  - Crypto-currencies, drugs, and pornography
  - Mostly English content

Ethical considerations

- Content retrieved in an automatic way
- Only text, images were discarded to avoid the possible commission of a crime

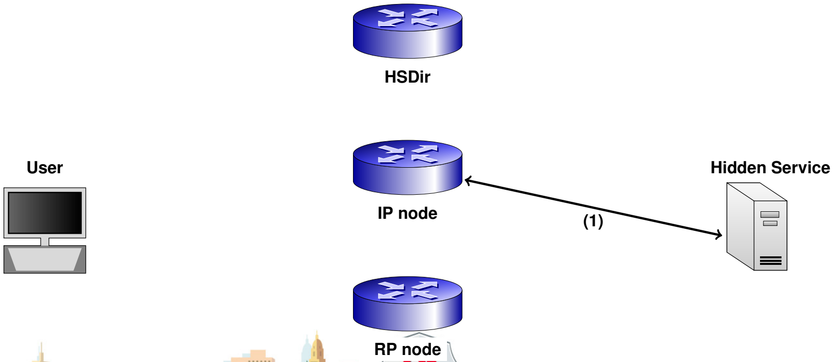
Introduction

How does a connection to a Hidden Service work?



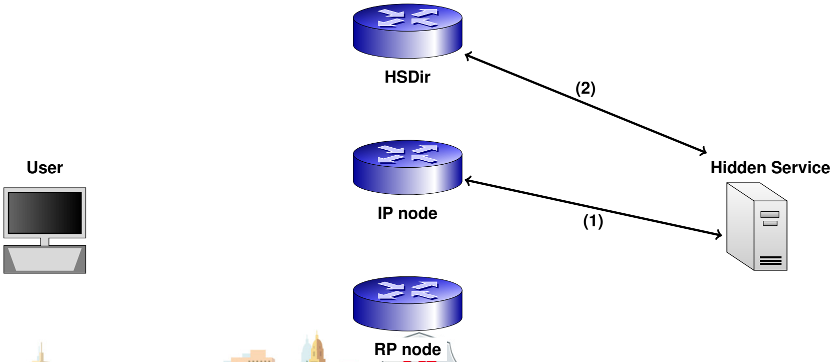
Introduction

How does a connection to a Hidden Service work?



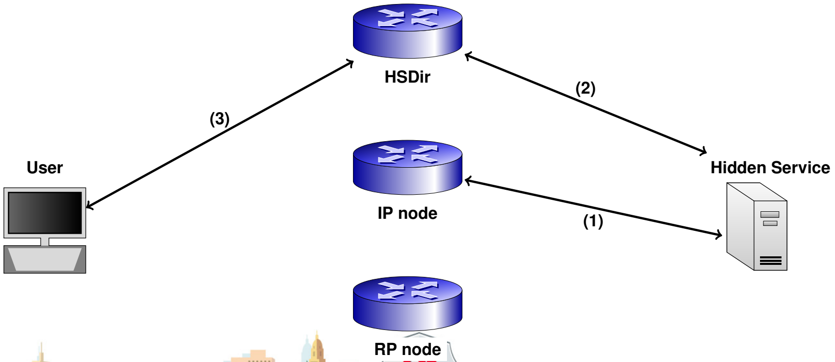
Introduction

How does a connection to a Hidden Service work?



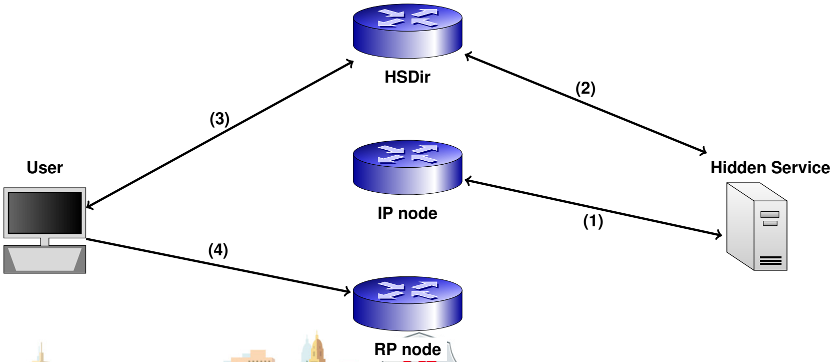
Introduction

How does a connection to a Hidden Service work?



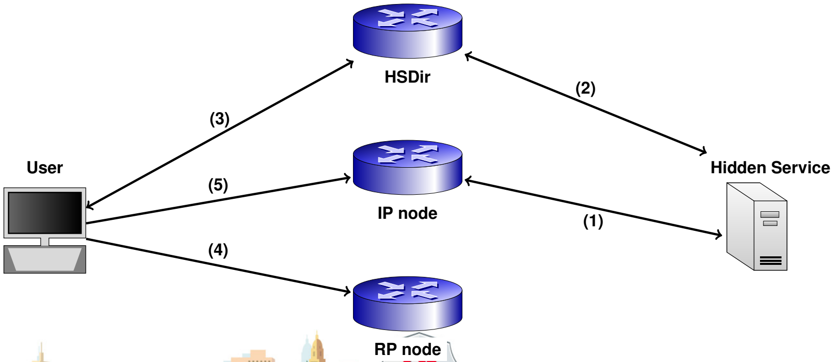
Introduction

How does a connection to a Hidden Service work?



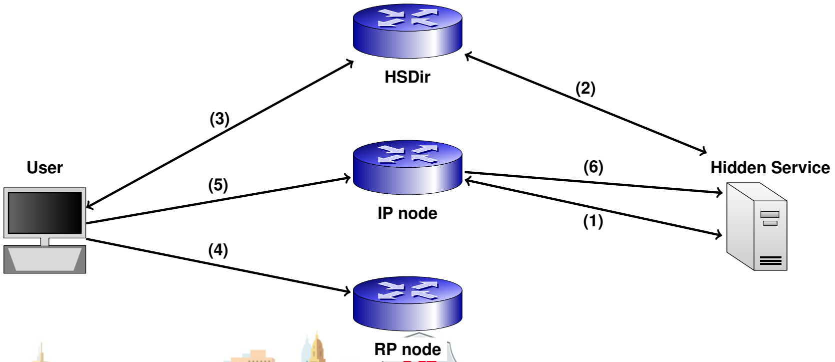
Introduction

How does a connection to a Hidden Service work?



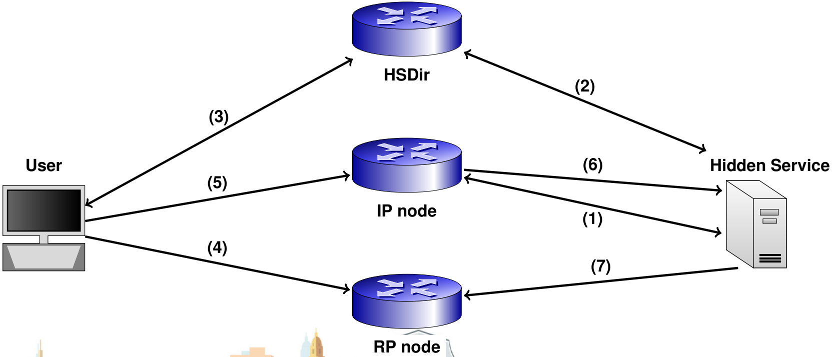
Introduction

How does a connection to a Hidden Service work?



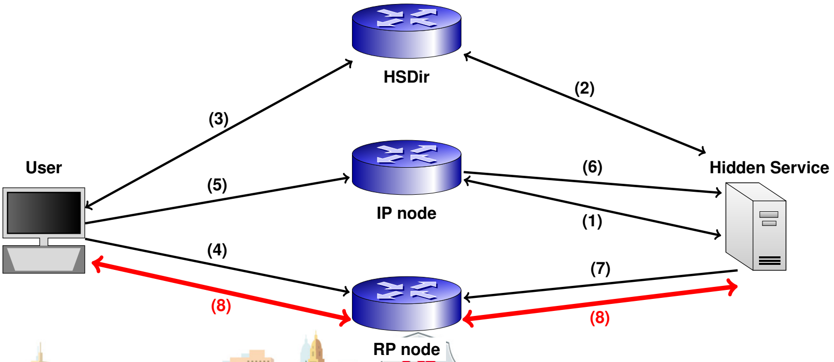
Introduction

How does a connection to a Hidden Service work?

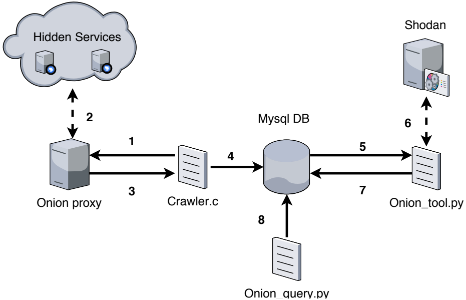


Introduction

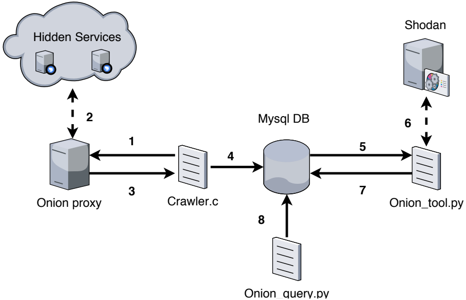
How does a connection to a Hidden Service work?



Description of TorHSScanner



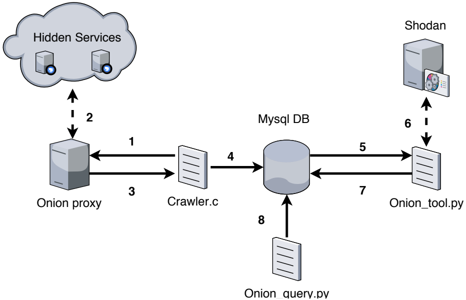
Description of TorHSScanner



1 Collection of Hidden Service addresses (through a crawler)

- **HTTP + HTTPS requests**
- When successful, **HTML of the land page is retrieved and stored**

Description of TorHSScanner



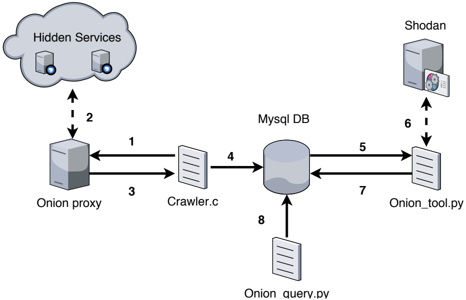
1 Collection of Hidden Service addresses (through a crawler)

- **HTTP + HTTPS requests**
- When successful, **HTML of the land page is retrieved and stored**

2 Deanonymization

- **Internet metadata provided by Shodan**
- **Greedy algorithm to find similarities**

Description of TorHSScanner



1 Collection of Hidden Service addresses (through a crawler)

- **HTTP + HTTPS requests**
- When successful, **HTML of the land page is retrieved and stored**

2 Deanonymization

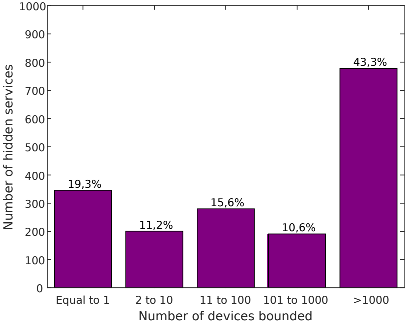
- **Internet metadata provided by Shodan**
- **Greedy algorithm to find similarities**

3 Categorization

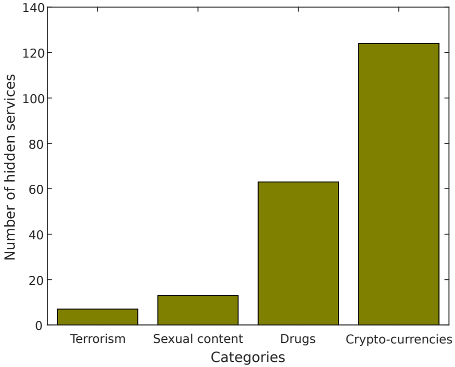
- **Categories**: drugs, sexual content, crypto-currencies, terrorism
- Every category has a **bag of words** (marijuana, porn, IED, etc.)
- **Natural language processing** libraries (in particular, NTLK)

Experimental Results

- **17328 .onion addresses collected** (≈ 15 % of official ones)
- **Successful HTTP/HTTPS connections to** 1796 **addresses**
  - Timeout set to 30 seconds
  - A hidden service can be configured to only allow access through access password
  - Many HSs change the domain frequently to avoid fingerprinting or other deanonymization techniques
- 30 % **of those bounded to (less than or equal to) 10 devices**



Experimental Results

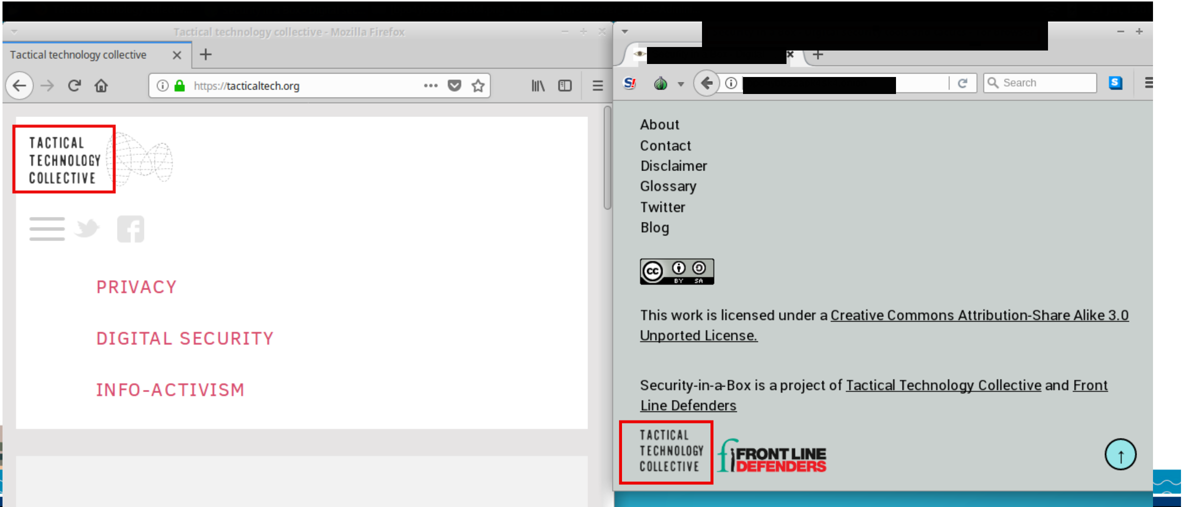


Idiom	# HS	Idiom	# HS
English	1313	Romanian	4
German	54	Turkish	4
Danish	38	Welsh	3
Portuguese	33	Slovak	3
Spanish	25	Swedish	3
French	19	Swahili	3
Italian	8	Tagalog	3
Norweigan	8	Vietnamese	3
Afrikaans	7	Indonesian	2
Dutch	7	Bulgarian	1
Somali	7	Estonian	1
Finish	6	Lithuanian	1
Polish	5	Albanian	1
Catalan	4	Unknown	229

- **28 different languages** (only 55 supported by NTLK)
  - **Most common: English** (1314 hidden services)
  - Spanish appears in only 25 services

Experimental Results

Deanonymization examples – same organization



[redacted]

Tactical technology collective - Mozilla Firefox

Tactical technology collective

https://tacticaltech.org

PRIVACY

DIGITAL SECURITY

INFO-ACTIVISM

[redacted]

[redacted]

[redacted]

Search

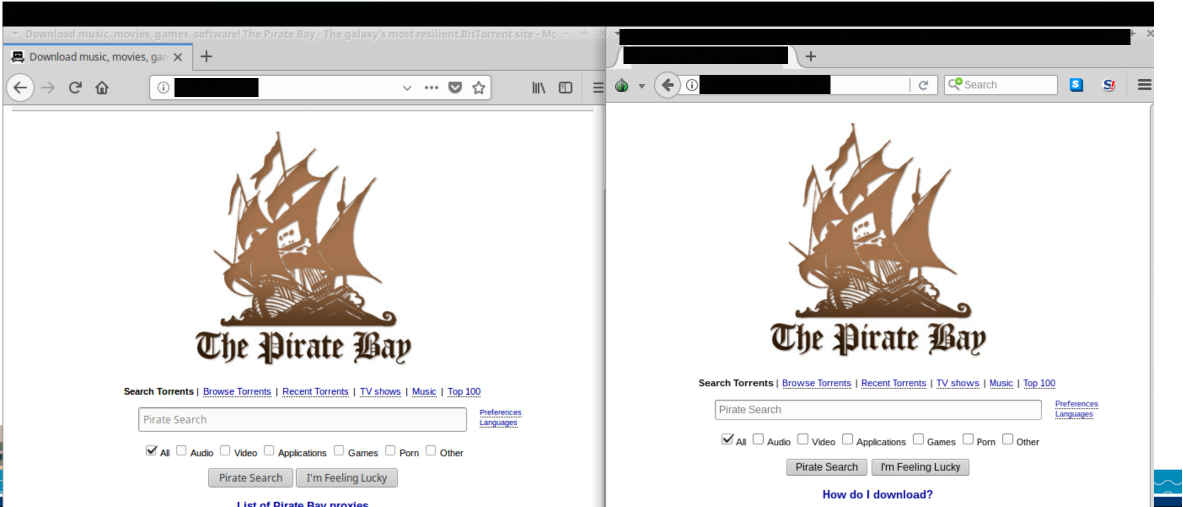
About
Contact
Disclaimer
Glossary
Twitter
Blog

This work is licensed under a Creative Commons Attribution-Share Alike 3.0 Unported License.

Security-in-a-Box is a project of Tactical Technology Collective and Front Line Defenders

Experimental Results

Deanonymization examples – same content



[redacted]

Download music, movies, games, software! The Pirate Bay - The galaxy's most resilient BitTorrent site - Mc

Download music, movies, gar

[redacted]

Search Torrents | Browse Torrents | Recent Torrents | TV shows | Music | Top 100

Pirate Search

Preferences
Languages

☑ All □ Audio □ Video □ Applications □ Games □ Porn □ Other

Pirate Search | I'm Feeling Lucky

List of Pirate Bay proxies

[redacted]

[redacted]

[redacted]

Search

Search Torrents | Browse Torrents | Recent Torrents | TV shows | Music | Top 100

Pirate Search

Preferences
Languages

☑ All □ Audio □ Video □ Applications □ Games □ Porn □ Other

Pirate Search | I'm Feeling Lucky

How do I download?

Agenda

1 Introduction

2 Modeling the Tor Network

3 Towards the Deanonymization of Tor Hidden Services

4 Related Work

5 Conclusions

Related Work

- **Deanonymization attacks through a direct participation in the virtual circuit**
- **Web traffic-based pattern analysis** (website fingerprinting attacks)
- **Exhaustive classification of Tor hidden services** in [OS-IET.IFS-16]
  - **On Tor version 2. We are using Tor version 3** (most secure for hidden services) – *that kind of attack is no longer working :(*
- 6426 addresses collected in [M-PhDThesis-16]. Connection was made to 1974 and **deanonymization success rate was** 5 %

OS-IET.IFS-16 Owen G, Savage N. *Empirical analysis of Tor Hidden Services*. IET Information Security. 2016;10(3):113–118.

M-PhDThesis-16 Matic S. *Active Techniques for Revealing and Analyzing the Security of Hidden Servers*. Università degli Studi di Milano, Milan, Italy; 2016

Agenda

1 Introduction

2 Modeling the Tor Network

3 Towards the Deanonymization of Tor Hidden Services

4 Related Work

5 Conclusions

Conclusions

Regarding the modeling of the Tor network

- **UML diagrams covering different aspects of the Tor network**
  - **Elements of the network** (nodes, messages)
  - **Communication**
  - **Node behavior**
- **This is the first step to perform a model-based verification**

Conclusions

Regarding the modeling of the Tor network

- **UML diagrams covering different aspects of the Tor network**
  - **Elements of the network** (nodes, messages)
  - **Communication**
  - **Node behavior**
- **This is the first step to perform a model-based verification**

Future work

- **Perform a model-based verification**
- **Transform to formal models to evaluate the behavior** (i.e., Petri nets)

Conclusions

Regarding TorHSScanner

- **Development of a tool to collect hidden service addresses and deanonymize them**
  - **Fingerprinting based on metadata of HTTP and HTTPS headers**
  - **HTML of the land page retrieved and analyzed using NLP toolkits**
- **Connection established to 1796 hidden service addresses** (17328 collected)
  - **Good success rate** (30 % bounded to less than 10 Internet systems)
  - Prevalence of **drug dealing and crypto-currencies** services
  - **English language is mostly common**

Conclusions

Regarding `TorHSScanner`

- **Development of a tool to collect hidden service addresses and deanonymize them**
  - **Fingerprinting based on metadata of HTTP and HTTPS headers**
  - **HTML of the land page retrieved and analyzed using NLP toolkits**
- **Connection established to 1796 hidden service addresses** (17328 collected)
  - **Good success rate** (30 % bounded to less than 10 Internet systems)
  - Prevalence of **drug dealing and crypto-currencies** services
  - **English language is mostly common**

Future improvements

- **Collection of HS addresses** (investigate other methods)
- **Deanonymization subsystem** (not Shodan)
- **Better categorization** (current method is not so good; semantic web?)

GRACIAS