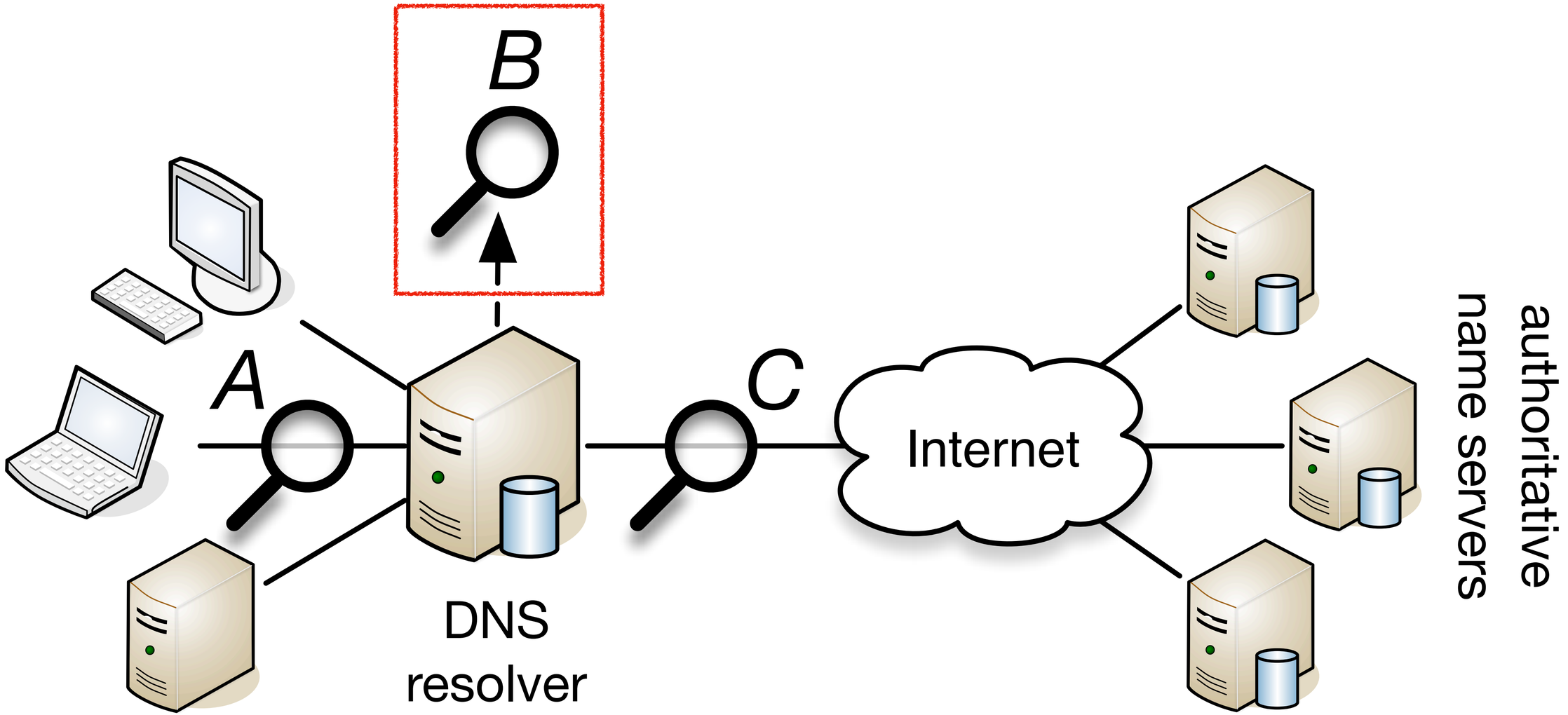
Privacy-Conscious Threat Detection Using DNSBloom

Roland van Rijswijk-Deij, Gijs Rijnders,
Matthijs Bomhoff and Luca Allodi

Introduction

- **Privacy of DNS traffic** between client and resolver currently **has a lot of attention** in the IETF, e.g.:
  - **DPRIVE** working group in the IETF, standardised **DNS-over-TLS**
  - **Deployment of DNS-over-TLS** by e.g. 1.1.1.1, 8.8.8.8, 9.9.9.9 and others, default support in e.g. Android P
  - **The buzz around** DNS-over-HTTPS (**DoH**)
- But the **focus** of these **is** mostly **on privacy of traffic in-flight**

Elephant in the Room



Resolver Operators

- Resolver **operators can still observe and collect** DNS query **traffic**
- And they have **legitimate reasons** to do so
- For example:
  - To **detect indicators of compromise** in DNS traffic (e.g. in enterprise networks)
  - To **monitor threats in large user bases** (e.g. Quad9)

Privacy-Conscious Monitoring

- We asked ourselves:
  "How can we detect if certain DNS queries were performed, while respecting the privacy of users?"

- Last year, **we developed a potential solution** for this problem:
  use of so-called **Bloom Filters**

- **Working prototype** available in **open source**

- **Tested in production** at SURFnet (national research network)

Bloom Filters

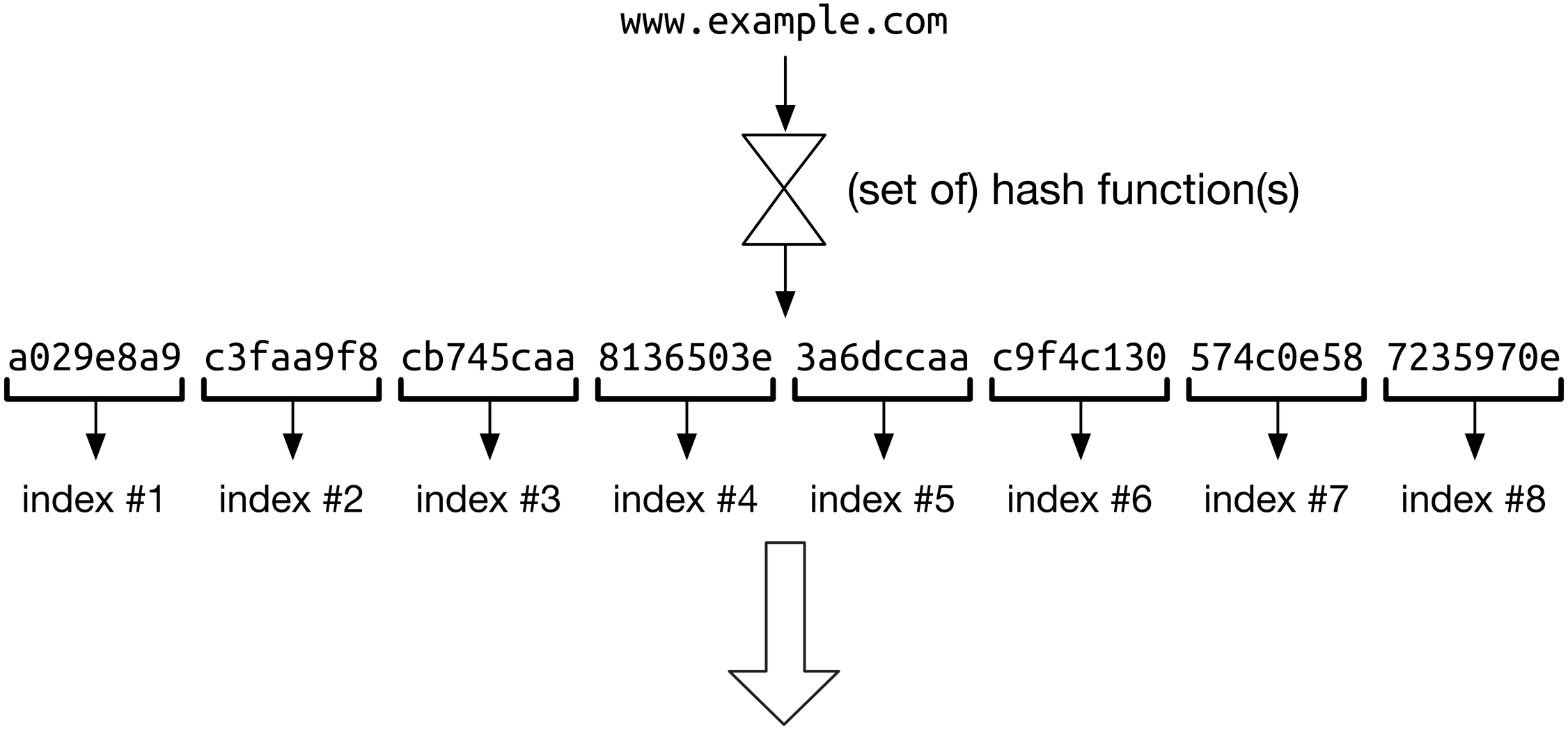
- **Developed in the 1970s** to speed up database lookups
- **Highly efficient**, insertion and lookup are ~$\mathcal{O}(1)$
- **Bloom Filters are** like **a set with a probabilistic membership test**
- For a given Bloom Filter B and an element n, we can test the following:

$n \in B?$

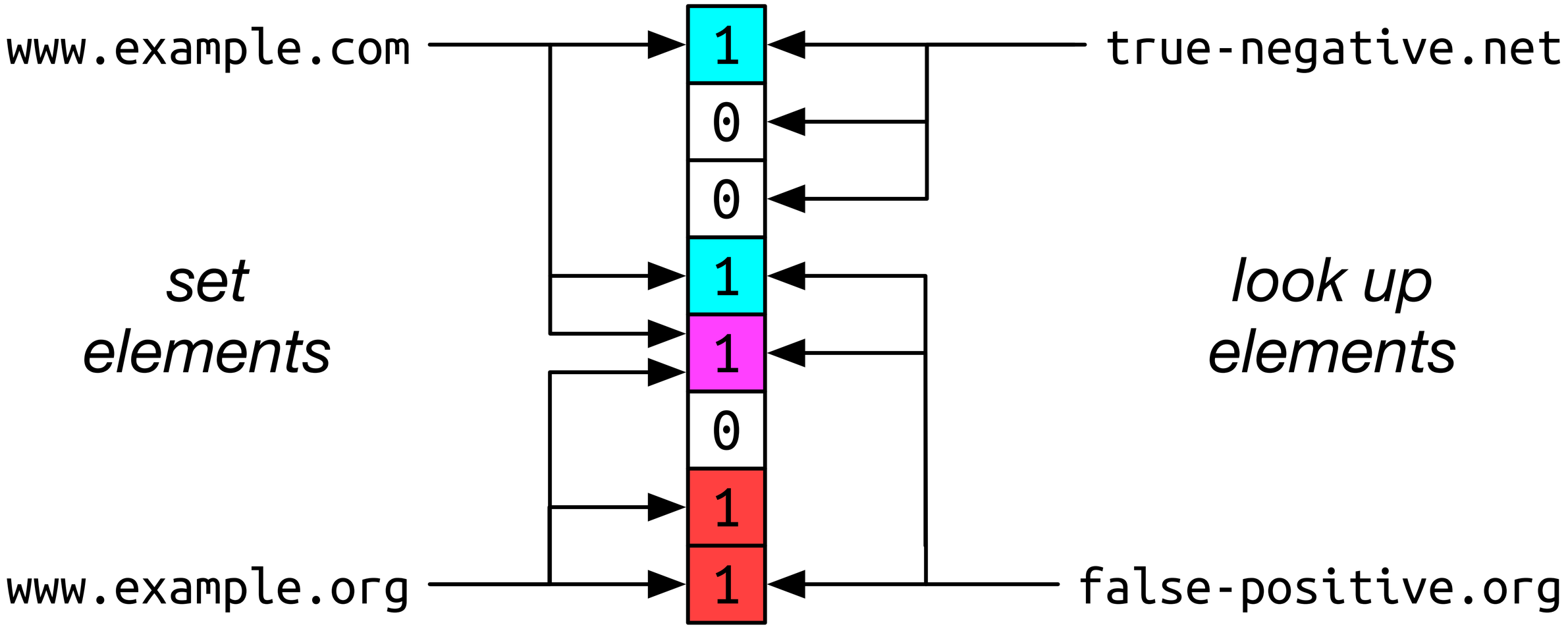
no → n is **guaranteed not** to be **in** B

yes → n is **highly likely in** B, with a **small probability** p_{ε} of this being **a false positive**

Bloom Filters



Bloom Filters



Bloom Filter Parameters

- **Tradeoff between** (low) **false positive rate and** a reasonable **filter size**

- Parameters:
  - Number of hash functions k → number of indices
  - Size of bit array m
  - Expected number of distinct elements n

- The formula below approximates the probability of a false positive p_ε:

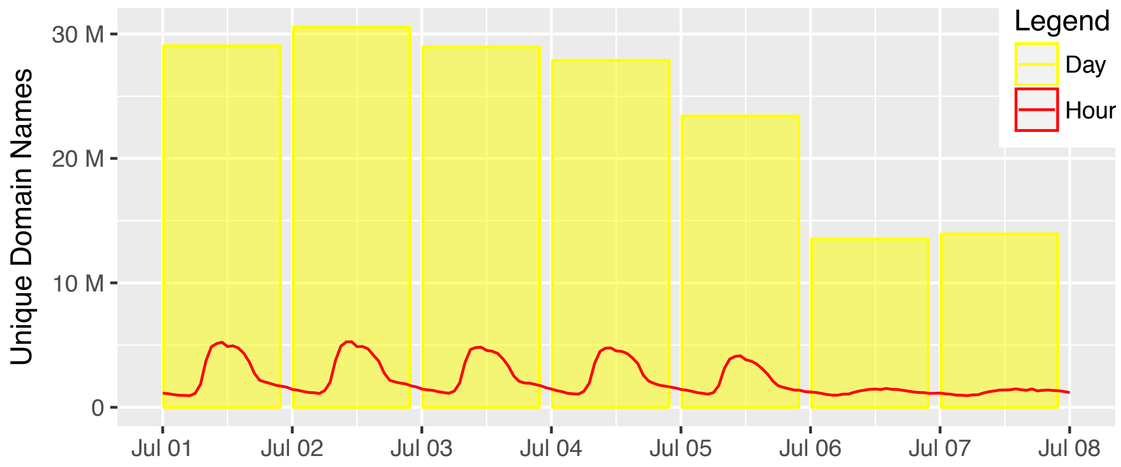
$$p_\epsilon \approx \left(1 - e^{-\frac{kn}{m}}\right)^k$$

The Idea

- Insert all queries from clients of a resolver into a Bloom Filter
- Then, we can check *if* a name was queried for, but *not by whom* and also *not exactly when*; this is sufficient for network-level threat monitoring
- **Privacy properties** of Bloom Filters:
  - **Non-enumerable**
  - By **mixing** queries from many **users** in a single filter, tracking becomes harder
  - Due to **mathematical properties** of Bloom Filters, we can **combine different filters**, so we can **further aggregate data over time** (making it even harder to track user)

Tuning Filter Sizes

- Our prototype creates **Bloom Filters per hour**
- Filters tuned to allow **aggregation** of 24 hourly filters **to one day** at maximum $p_\varepsilon = 0.001$ (or one in a thousand)
- Prototype supports "**auto tuning**", in which it first gathers statistics on query traffic for some time and then suggests parameters
- Future extension: continuous auto tuning



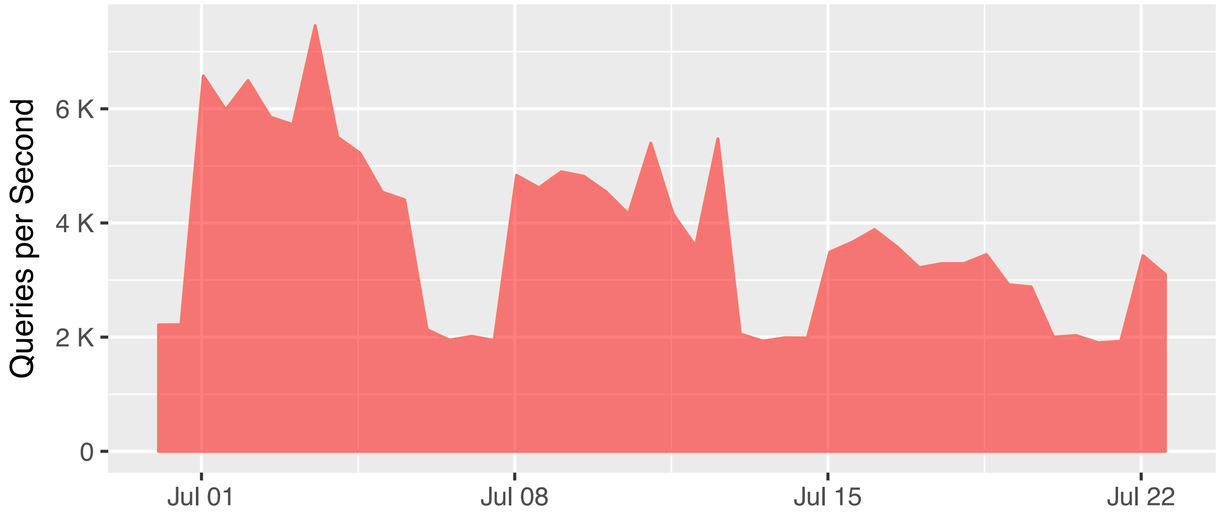
Filter Input

- Our prototype implementation allows distinguishing between organisations on a network (specific requirement of test environment)
- We insert the full query name, the second level domain and every label except for the public suffix
- Inserted twice, generic and with organisation name prepended

Description	Value	Description	Value
FQDN	`evil.domain.com`	FQDN	`Organisation A@evil.domain.com`
2nd-level	`domain.com`	2nd-level	`Organisation A@domain.com`
Label 1	`evil`	Label 1	`Organisation A@evil`
Label 2	`domain`	Label 2	`Organisation A@domain`

Real-World Tests

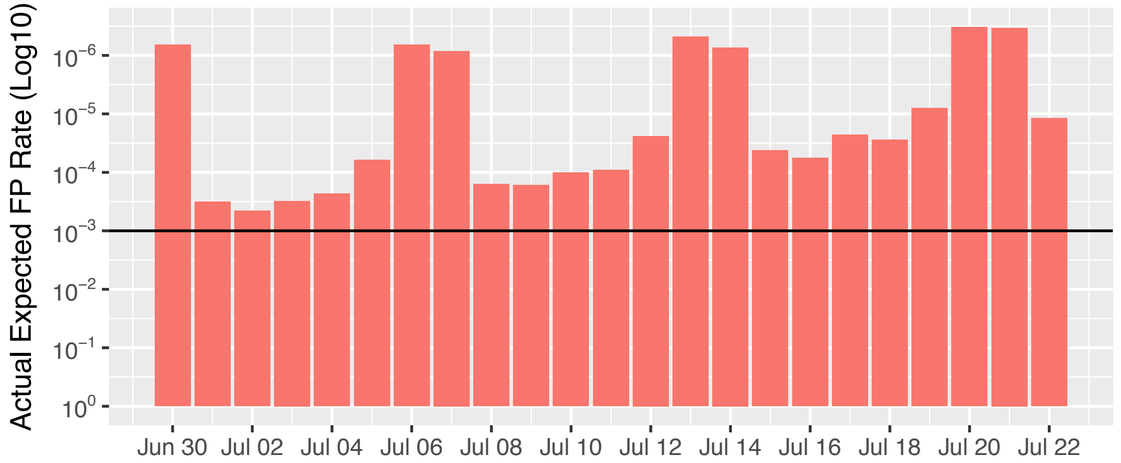
- We **tested** this **for three weeks** on busy DNS resolvers at SURFnet
- We studied **three use cases**:
  - Detection of so-called "**Booters**"
  - Hits on **e-mail blacklists**
  - Hits of high-value indicators-of-compromise for the so-called **National Detection Network**



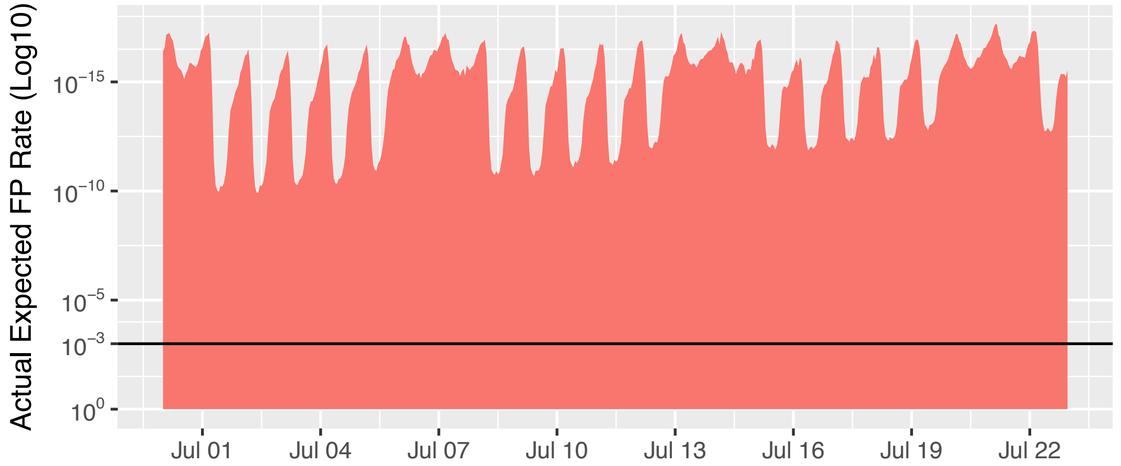
Predicted vs. Actual FPR

- We ran **auto tuning for a week** and chose the following filter **parameters**:
  $k = 10$
  $m \approx 491$ Mbits (±59MB)
- **Goal: daily FPR** of **one in a thousand**
- Of course, we had to **estimate** the **number of elements inserted** (n)
- After using a filter, we can calculate the actual false positive rate (without knowing n):

$$p_a = \left(\frac{s}{m}\right)^k$$



24h actual false positive rate (higher = better)

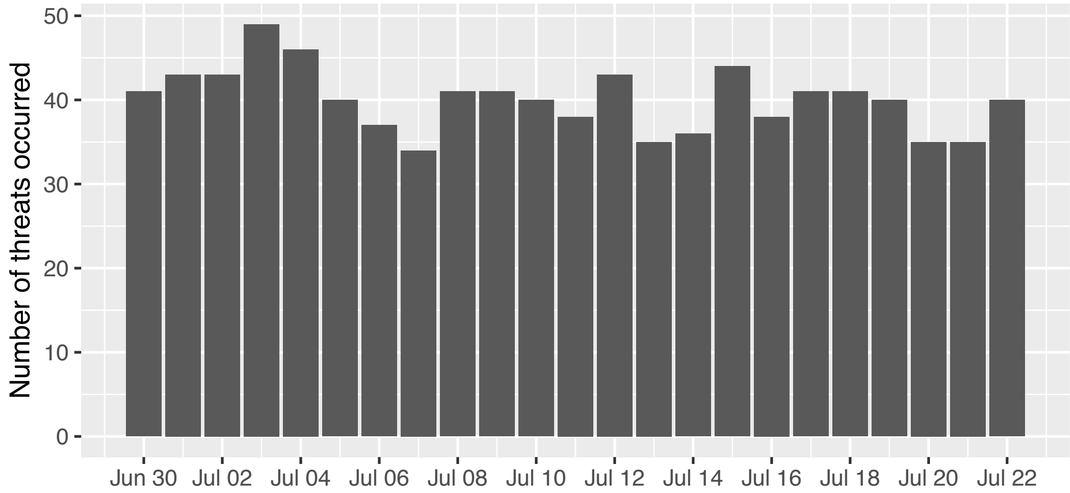


1h actual false positive rate (higher = better)

National Detection Network

- NDN is **managed by** the Dutch National Cyber Security Centre (**NCSC**) and is supposed to have **"high value" indicators-of-compromise** (from **e.g. intelligence services**)

- **SURFnet could** previously **not monitor for threats** reported in NDN because monitoring **DNS traffic** was considered **too privacy sensitive**



- **With Bloom Filter approach** it was **now possible**, and we **found actual compromises** (e.g. WannaCry infected machine)

Targeted Logging

- **For** the WannaCry **infection**, we could now **apply targeted logging**
- Rather than monitoring everyone's queries, **only look for a specific query from a specific subnet**
- **Much less invasive for users!**

Other Benefits

- **No personal data** is **stored**, so **data can be retained** for longer periods
  → **Enables historical lookups** (did this new threat occur in the past?)

- Bloom Filters could be **shared with third parties**, or you could allow lookups from third parties
  → Check for **occurrence of threats in multiple networks**
  → **Allow researchers access**

- Properties of Bloom Filters are similar to HyperLogLog, so it is possible to do **cardinality estimates** of the number of items (**distinct queries**)

Conclusions

- **Prototype code** has been **released as open source** https://github.com/SURFnet/honas
- **SURFnet** is **planning to take** this **into production**
- **Future integration in** NLnet Labs **open source software** to make this approach more widely available and **easy to deploy**
- **Proof that security and privacy *can* go hand in hand!**

Privacy-Conscious Threat Intelligence Using DNSBLOOM

Roland van Rijswijk-Deij*†, Gijs Rijnders‡§, Matthijs Bomhoff§ and Luca Allodi‡
*University of Twente, †NLnet Labs, §Tesorion, ‡Eindhoven University of Technology
r.m.vanrijswijk@utwente.nl, gijs.rijnders@tesorion.nl, matthijs.bomhoff@tesorion.nl, l.allodi@tue.nl

***Abstract*—The Domain Name System (DNS) is an essential component of every interaction on the Internet. DNS translates human-readable names into machine readable IP addresses. Conversely, DNS requests provide a wealth of information about what goes on in the network. Malicious activity – such as phishing, malware and botnets – also makes use of the DNS. Thus, monitoring DNS traffic is essential for the security team's toolbox. Yet because DNS is so essential to Internet services, tracking DNS is also highly privacy-invasive, as what domain names a user requests reveals their Internet use. Therefore, in an age of comprehensive privacy legislation, such as Europe's GDPR, simply logging every DNS request is not acceptable.**

In this paper we present DNSBLOOM, a system that uses Bloom Filters as a privacy-enhancing technology to store DNS requests. Bloom Filters act as a probabilistic set, where a membership test either returns probable membership (with a small false positive probability), or certain non-membership. Because Bloom Filters do not store original information, and because DNSBLOOM aggregates queries from multiple users over fixed time periods, the system offers strong privacy guarantees while enabling security professionals to check with a high degree of confidence whether certain DNS queries associated with malicious activity have occurred. We validate DNSBLOOM through three case studies performed on the production DNS infrastructure of a major global research network, and release a working prototype, that integrates with popular DNS resolvers, in open source.

***Index Terms*—DNS; privacy; measurement; GDPR; threat detection; indicator-of-compromise**

I. INTRODUCTION

In modern networks, there is a constant arms race between network managers and miscreants that want to infiltrate the network, to deploy botnets, to infect users with malware or to phish their credentials. Consequently, network security professionals need to have a well-stocked toolbox to combat such adversaries. A well-known approach to threat detection is to monitor Domain Name System (DNS) queries. The DNS fulfills a key role for Internet services: it maps human-readable names to machine-readable IP addresses. Because DNS is so essential, malicious activity on a network oftentimes relies on the DNS in some way. This can be either just to map names to addresses, e.g., for URLs included in phishing e-mails, or more active abuse of the DNS, for instance as a command-and-control (C&C) channel for botnets.

A major problem with monitoring DNS queries on a network is that this is also extremely privacy-invasive [1], [2]. Because almost all network services rely on the DNS in some way, recording what DNS queries a user performs is highly revealing of their Internet use. In the age of ever stricter privacy legislation – think, for example, of Europe's General Data Protection Regulation (GDPR) [3] – simply recording all DNS traffic on a network is not considered *proportional* to the goal of safeguarding network security. Given, however, how valuable DNS query logs can be for network security, the following question is worth asking: *Can we track information about DNS queries without compromising on user privacy?*

In this paper we present DNSBLOOM, a system that uses Bloom Filters [4] as a privacy enhancing technology to track DNS queries. Bloom Filters were invented in the 1970s as a time- and space-efficient means to index databases. They act as a probabilistic set, where a membership test either confirms *certain* non-membership, or indicates *probable* membership with a low probability of false positives. Bloom Filters rely on hash functions to store information; as such, they never store the original information. DNSBLOOM leverages this property to protect user privacy while retaining useful detection properties. In essence, when using DNSBLOOM, a security professional can ask *if* a specific query for a known (malicious) domain name has occurred, but cannot obtain a set of *all* queries that occurred in the network. While this does not allow for real-time monitoring of threats, it does allow for tactical and strategic assessment of threats on a network: upon observation of threats (known as *indicators-of-compromise* – IoCs) using DNSBLOOM, security professionals can decide to deploy targeted monitoring for specific threats, thus achieving a proportional (e.g., in the sense of the GDPR) collection of data. Moreover, DNSBLOOM allows operators to keep track of DNS queries over time – in a privacy-conscious manner – and to look back in time to see if emerging threats have already occurred in their network.

To demonstrate its practical value, we validate the use of DNSBLOOM in three real-world scenarios at a major global research network. Furthermore, we implement a working prototype that seamlessly integrates with all major open source DNS resolver implementations. This prototype is released in open source, to foster reproducibility and future research.

Paper organization — the remainder of this paper is organised as follows. Section II provides background information on Bloom Filters and IoCs. Section III introduces the approach behind DNSBLOOM. In Section IV, we report on the evaluation of the DNSBLOOM prototype. Section V reflects on the results of our validation, and Section VI, discusses conclusions and provides an outlook on future research.

978-3-903176-15-7 © 2019 IFIP

Thank you! Questions?

nl.linkedin.com/in/rolandvanrijswijk

@reseauxsansfil

roland@nlnetlabs.nl

Paper available as open access (thanks to IFIP):
http://dl.ifip.org/db/conf/im/im2019/189282.pdf