Let a Thousand Filters Bloom

DNS-Based Threat Monitoring That Respects User Privacy

Roland van Rijswijk-Deij
SURFnet bv
roland.vanrijswijk@surfnet.nl

Matthijs Bomhoff
Quarantainenet B.V.
matthijs@quarantainenet.nl

Ralph Dolmans
NLnet Labs Foundation
ralph@nlnetlabs.nl

Keywords

DNS, privacy, security, privacy enhancing technology, network monitoring

1. INTRODUCTION

The one phrase that consistently comes to mind when describing the Internet in 2017 is: *"Here be dragons"*. The users of the Internet, including the users of our NREN networks face increasing threats. From malicious actors out to infect them with malware and ransomware, or to phish their credentials, to nation state actors out to intercept their traffic in efforts to implement 1984-style blanket surveillance. To counteract these threats, the Internet standards community is adapting existing protocols and developing new ones to increase user security and privacy. One issue that keeps coming up in these efforts is the hard-to-avoid choice between privacy and security.

A good illustration of this problem are the efforts to improve the privacy of users of the Domain Name System. The DNS provides a vital role on the Internet, mapping human readable names to machine readable addresses. For most end users, the process of DNS resolving is outsourced to a DNS resolver upstream on their network. Many NRENs provide DNS resolver services to their constituency for this purpose. The classic DNS protocol has no provisions for confidentiality. This makes users vulnerable to profiling; knowing which names a user tries to resolve can be very revealing about their Internet use. For this reason, the IETF started the DPRIVE working group, whose goal is to improve the privacy of the DNS. A concrete outcome of the work in this group is the DNS-over-TLS protocol [1], which provides confidentiality for the communication between a client and a DNS resolver. While this provides privacy for users, it has disadvantages. Many monitoring systems that try to detect threats to users rely on monitoring DNS traffic for queries to suspicious domains that are, e.g., associated with malware or botnets. If DNS traffic is encrypted, such systems can no longer inspect traffic. An obvious solution to this problem is to extend DNS resolver software to inspect queries there, but this just changes the place in which a user's privacy is infringed from traffic inspection to the DNS resolver. In this paper, we discuss an alternative approach to solving this problem that allows for DNS query inspection while at the same time protecting user privacy. Our main contributions are that we:

- Introduce a new approach to DNS-based threat detection that uses privacy-enhancing technology to protect user privacy;
- Demonstrate how this technology performs in practice on an NREN network;
- Release the software into the public domain, with a concrete implementation in a popular open source DNS resolver package.

2. GOALS AND APPROACH

Like many network operators, SURFnet wants to get more insight into the threats that our users are facing. In order to do this, we want to be able to identify so-called *Indicators of Compromise* (IOCs). One of the ways to do this is to monitor for DNS queries to the DNS resolvers we operate for our constituency. If we look at this problem abstractly, then our goal is to identify if certain DNS names, associated with IOCs, are queried for by clients of our resolvers. Ideally, we want to be able to identify queries from specific subnets (e.g. the subnet of a specific institution), so we can notify the local incident responders. Inspecting DNS traffic significantly infringes user privacy, therefore, our second goal is that we want to do this in a manner that respects the privacy of individual users. In other words: we do not want to store information about, or be able to monitor the DNS queries of an individual user.

Our approach to achieving these two goals is to use so-called *Bloom filters* [2] as a privacy-enhancing technology. Bloom filters can best be explained as a statistical way to test for set membership. Items that are added to a Bloom filter are run through a set of hash functions, and the output of these functions are used to set bits in a bit array. The contents of this bit array are then used to test set membership. In a Bloom filter, non-membership can be guaranteed. In other words, if a lookup returns "element is not a member", then this result is certain. Membership proof, on the other hand, is probabilistic. That is: depending on certain configurable parameters, a lookup with a positive result means that there is a high probability that the element is a member of the set of elements added to the Bloom filter. Typically, filters are configured to give a high-confidence ($\geq 95\%$) answer.

In the context of our goals, Bloom filters offer the following benefits:

- The filter does not store original query names; conversely, what queries were performed cannot be trivially recovered from the filter;
- A lookup can only be performed if you know the precise name your are looking for;
- By mixing queries from a sufficient number of users in a single filter, it becomes practically impossible to check if a single user performed a certain query;
- Bloom filters provide a memory-efficient way to track large numbers of queries;
- The state of Bloom filters can be stored to make it possible to perform historic lookups;
- The state of different Bloom filters can be combined, which, e.g., allows us to combine Bloom filters from different DNS resolvers with different user populations.

3. IMPLEMENTATION

The use of Bloom filters to keep track of DNS queries was suggested and implemented by Quarantainenet[1], a Dutch company that specialises in network management and security. SURFnet has partnered with Quarantainenet to make their Bloom filter implementation available as open source software. In addition to this, we wanted to lower the barrier of deploying this solution in practice. Therefore, SURFnet also partnered with NLnet Labs, who develop the popular open source Unbound DNS resolver[2], to integrate the Bloom filter solution in Unbound. This has the added advantage that this solution becomes widely available to network operators in an easy to deploy manner.

4. TEST DEPLOYMENT

In the first half of 2018 SURFnet will deploy and test this solution on the three DNS resolvers it operates for its constituency. During this test phase, we will experiment with different ways to assign users to Bloom filters, in order to test different detection scenarios. After this test phase, we plan to start using this technique to assist our incident response team in tracking IOCs.

5. CONCLUSIONS AND FUTURE WORK

We have shown that it is possible to monitor networks for threats while at the same time respecting the privacy of users on the network. This shows that privacy and security do not have to be mutually exclusive, as is often assumed. In the context of stricter controls on what network operators can do that stem from, e.g., the European GDPR, this is a promising result. And by making the software SURFnet uses for this goal available in open source, we contribute to a safer *and* more privacy-friendly Internet for everyone.

In the near future, our most important goal is to perform more extensive experiments with the solution presented in this paper, and to develop guidelines based on our tests for deploying this technology in diverse networks. One of the goals of presenting this work at TNC is to encourage other NRENs to deploy this system. In addition to this, we plan to investigate if we can make this technology available in other open source DNS resolver implementations, such as BIND, PowerDNS Recursor and Knot Resolver.

6. REFERENCES

[1] Z. Hu, L. Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul Hoffman. RFC 7858 - Specification for DNS Over Transport Layer Security (TLS). https://tools.ietf.org/html/rfc7858, 2016.

[2] Burton H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communications of The ACM*, 13(7):422–426, July 1970.

[1] `https://quarantainenet.nl` (note: site is in Dutch)

[2] `https://unbound.net/`