The authors present a control plane for operators of top-level domains (TLDs) in the DNS, such as“.org” and “.nl,” that enables them to increase the security and stability of their TLD by taking on the role of a threat intelligence provider. Our control plane is a novel system that extends a TLD operator’s traditional services and detects potential threats in the TLD by continuously analyzing the TLD operator’s two key datasets: the typically large amounts of DNS traffic that it handles and its database of registered domain names. The control plane shares information on discovered threats with other players in the TLD’s ecosystem and can also use it to dynamically scale the TLD operator’s DNS infrastructure. The control plane builds on a set of open source modules that we have developed on top of a Hadoop-based data storage cluster. These enable, for example, TLD operators to run and develop threat detectors and to easily import their DNS traffic into the control plane. Our control plane uses policies to protect the privacy of TLD users and is based on our operational experience of running .nl TLD (Netherlands), which we are also using as the use case for our implementation.

INTRODUCTION

Since their inception, domain names have been used as a simple identification label for hosts, services, applications, and networks on the Internet (RFC 1034). Until the mid-1980s, the mappings from domain names to IP addresses were distributed as text files (HOSTS.TXT) via ftp to the relatively small number of hosts that were connected to the Internet at that time. The Domain Name System (DNS) (RFC 1034) replaced this mechanism to provide domain name to IP address mappings in a scalable way and has become a critical part of the Internet infrastructure.

The DNS uses a hierarchical namespace and a tree-like structure in which each level uses so-called authoritative name servers to provide pointers to the next lower level. As an example, consider a user tying to reach the website www.example.nl (Fig. 1). The user’s computer first connects to a resolver, which is a recursive name server that interacts with authoritative name servers on behalf of the user and is usually located in the network of the user’s Internet access provider. The resolver obtains a reference to the “.nl” namespace from the root name servers, then a reference to “example.nl” from the .nl name servers, and finally the reference to “www.example.nl” from the name server of example.nl. This last name server knows the requested IP address, which the resolver returns to the user, allowing its browser to reach www.example.nl.

The second level of the DNS namespace currently contains over 1300 top-level domains (TLDs), classified into country code TLDs (e.g.,“.nl” and “.br”), generic TLDs (e.g.,“.com” and “.org”), and new generic TLDs such as “amsterdam” and “.shop.” The operators of these TLDs manage the TLD’s authoritative name servers and the database of all registered second-level domain names (usually of the form [domain].[tld]). They regularly export the database contents to a so-called zone file, which is the input for the TLD’s authoritative DNS servers. The other levels in the DNS tree follow this same principle, as Fig. 1 illustrates.

A recent development is that some TLD operators have extended their traditional role as DNS operator to also take on the role of threat intelligence provider. They leverage the updates of their domain name database and the DNS traffic they handle on their name servers to detect potential threats in their TLD, such as phishing sites [1], distributed denial of service (DDoS) attacks on the DNS [2, 3], and sites that distribute malware. The underlying rationale is to protect the TLD’s users by making this threat information available to other players in the TLD, such as hosting and access providers, thus helping them to better fight these threats (collaborative security).

The contribution of our work is that we have developed and implemented a so-called control plane that enables TLD operators to become threat intelligence providers. The control plane is a novel system that extends a TLD operator’s traditional services (registration and DNS) to automatically derive potential threats from DNS traffic, database updates, and potentially other sources. Our control plane makes this threat information available to other players in the TLD and can also use it to dynamically scale the TLD operator’s DNS services. Together, these two functions increase the level of automation of operating a
TLD because threat detection and DNS reconfiguration are mostly manual and ad hoc tasks today.

Our control plane builds on several open source modules we have developed on top of a Hadoop-based data storage cluster. For instance, they enable TLD operators to detect phishing sites and to easily import their DNS traffic into the control plane. Our modules are currently being used by at least six TLD operators, including .ca (Canada) and .at (Austria). Our control plane uses policies to protect the privacy of TLD users and is based on our operational experience of running the .nl TLD (7th largest TLD, 5.6 million domain names). We are also using .nl as the use case for our implementation.

In this article, we focus on the design and principles of our control plane and refer the interested reader to our previous work for more technical details and extensive analysis.

We first provide an overview of infrastructure that a TLD operator typically manages. Next, we discuss the threats to which TLDs are exposed, the functions our control plane needs to mitigate them, and how we realized the control plane. We end with a discussion on related work, conclusions, and future work.

**TLD OPERATOR INFRASTRUCTURE**

A TLD operator traditionally manages the set of authoritative name servers for the TLD and the TLD’s registration database.

**AUTHORITATIVE NAME SERVERS**

Because TLD operators form the second highest level in the DNS naming hierarchy (Fig. 1), they typically use multiple layers of redundancy to provide their DNS services in a fault-tolerant way. For example, they replicate their name servers across multiple DNS services (e.g., ns1.dns.nl and ns2.dns.nl for the .nl TLD), use multiple types of name server software, and use IP anycast [3] to replicate their DNS services across sites. The advantage of IP anycast is that it also enables TLD operators to scale their DNS capacity to deal with an increasing DNS load and to reduce response times by placing machines closer to end users. IP anycast relies on the Internet’s inter-domain routing protocol (Border Gateway Protocol, BGP, RFC 4271) to route clients to the closest name server and is heavily used by the DNS root (11 of its 13 “letters” use anycast across more than 500 different locations [3]).

As an example, the DNS infrastructure for the .nl TLD consists of six unicast name servers and two anycast services. The anycast service is distributed across several dozens of sites, with one anycast service mostly co-located with large Dutch access providers (“local anycast”) and the others worldwide (through third parties). We use several different types of name server software for reasons of diversity, and changes to our infrastructure go through a tightly controlled change management process.

Four of our six unicast name servers together handle around 850 million DNS queries a day coming from approximately 1.3 million resolvers.\(^1\) This is a subset of the total amount of queries because resolvers use local caches to avoid having to completely walk the DNS tree for every lookup. This increases performance and DNS scalability, but implies that authoritative servers only receive part of the queries that a resolver receives from clients.

**REGISTRATION DATABASE**

A TLD operator’s registration database usually contains all the second-level domain names in a TLD, which are of the form [domain].[tld] (some TLD operators also allow for thirldlevel registrations, e.g., under .com.br). The TLD operator typically enables so-called registrars to register a domain name (or update or delete it) in the database on behalf of Internet users, which are called registrants. A registration corresponds to adding a leaf under a TLD in the DNS tree (Fig. 1).

Different registrars provide different registration interfaces, but the registrar-registry interface is often based on the Extensible Provisioning Protocol (EPP, RFC 5730). Registrars typically sell domain names in combination with hosting services.

As an example, the .nl registration database is synchronized across multiple sites, contains 5.6 million domains, and serves around 1500 domestic and international registrars. We offer both an EPP and a web-based interface, and generate and export the .nl zone file to our name servers every hour.

**THREATS**

The DNS and the domain names in a TLD are exposed to various threats. Some affect the services of a TLD operator, others those of other players within the TLD. We distinguish four types of threats in this article and refer to RFC 3833 for a more detailed description of DNS-related threats:

- **Zone file integrity violation:** These threats involve compromising the TLD zone file (cf. [16]), for instance, by stealing users’ or registrar credentials, allowing the attacker to change certain records in the zone file. This leads the authorita-\(^1\) http://stats.sidnlabs.nl

---

Figure 1. DNS naming hierarchy and DNS operators.
A TLD operator has two key datasets that it can use to detect threats: DNS authoritative traffic (incoming DNS queries for domains in the TLD’s zone) and the TLD’s domain registration database. The latter furthermore gives a TLD operator a real-time view on domain registration changes (creates, deletes, updates) across different registrars.

tive server to respond to queries with fraudulent answers, ultimately pointing the user to a malicious domain name.

Name server unavailability: This type of threat purposefully reduces the availability of name servers in the DNS, for instance, through a DDoS attack [2–4]. This results in name servers becoming unavailable or unstable (partial availability), which means that clients do not receive a response to their DNS request (in time) and are unable to reach the intended server.

DNS response integrity violation: Bad actors tamper with DNS responses, for instance, through man-in-the-middle attacks, DNS hijacking, or cache poisoning (RFC 3833). This results in a user being redirected to a malicious or unsolicited server. DNSSEC (RFC4035) detects this type of attack at the resolver.

Abuse: The DNS is being exposed to various sorts of abuse, such as phishing, malware distribution, and command-and-control botnet channels. While the malicious content is hosted outside the DNS, the DNS is misused to direct victims to such sites.

**DATA AND FUNCTIONS**

The goal of our control plane is to leverage the data that a TLD operator handles to detect potential threats in the TLD and to automatically reconfigure the TLD operator’s name servers. The analysis of the TLD operator’s data requires a third function, which is privacy protection.

**TLD OPERATOR DATA**

A TLD operator has two key datasets that it can use to detect threats: DNS authoritative traffic (incoming DNS queries for domains in the TLD’s zone) and the TLD’s domain registration database. The latter furthermore gives a TLD operator a real-time view on domain registration changes (creates, deletes, updates) across different registrars.

TLD operators can use these datasets to automatically detect patterns and suspicious behaviors in their zone. For example, the TLD operator would be able to detect spam campaigns based on bulk registrations, which has been reported on in [14]. It would also be able to detect phishing attacks based on unusual DNS traffic patterns for a domain that has just been registered (discussed later). TLD operators could furthermore cautiously carry out active measurements on all domain names in their zones and use this information to augment the threat detection logic.

While resolvers and DNS operators at lower levels in the DNS hierarchy would be able to carry out a similar analysis, they miss the real-time centralized view that a TLD operator has as a result of its position at the second-highest level in the DNS (Fig. 1). This makes it difficult for them to detect and correlate malicious domain names created through different registrars, such as the automatically generated domain names that botnets use.

The limitation of a TLD operator’s data is that it provides a “sampled” view of the DNS because resolvers cache queries [15]. Also, TLD operators are likely to gradually receive less DNS information because of QNAME minimization (RFC 7816), which is a recent DNS extension that reduces the amount of data in DNS queries to protect the privacy of users. QNAME minimization resolvers only put example.nl in the queries they send to TLD operators instead of www.example.nl, which is the fully qualified domain name (FQDN). The uptake of QNAME minimization is currently limited.

**THREAT DETECTION**

The purpose of threat detection is to automatically detect potential threats in a TLD, such as phishing domains and unavailability of DNS name servers. To accomplish this, the control plane needs to be able to quickly analyze large datasets covering a year or more of relatively high-volume DNS data. Speed is crucial to quickly detect and mitigate threats such as the appearance of phishing sites, which will affect fewer victims the sooner they are removed.

To accomplish this, the control plane needs to provide near-real-time response times when analyzing a TLD operator’s datasets, and needs to continuously store large volumes of DNS and other data. Data streaming warehouses (DSWs) [5] are designed with this in mind: they continuously digest incoming data, and use optimized file formats (columnar storage) and parallel processing to achieve near-real-time response times. DSWs can also easily be extended with extra nodes, enabling the control plane to increase its capacity when the TLD operator’s datasets grow. DSWs typically also provide an easy interface for data analysis, which eases application development and interaction with a human operator.

Our control plane’s DSW needs to be able to obtain the transport and IP-level information in DNS packets, which might be relevant, for example, to detect reflection attacks based on ICMP messages. The DSW should also introduce limited changes on the TLD operator’s name servers. This is essential because TLD-level name servers are high availability resources that are typically tightly managed. The format for importing DNS packets from name servers into the control plane should furthermore be widely used so that different TLD operators can easily implement our control plane irrespective of their particular name server setup (as discussed earlier).

We discuss our DSW later, and our threat detection modules and their performance after that.

**ON-DEMAND DNS RECONFIGURATION**

The purpose of on-demand DNS reconfiguration is to dynamically adapt the DNS anycast infrastructure of a TLD operator, for instance, to handle a DDoS threat (name server unavailability) as it occurs. TLD operators frequently use IP anycast because of its ability to handle stress situations [3] and because it allows them to easily scale their authoritative name server infrastructure.

By adapting we mean starting and stopping anycasted and virtualized DNS name servers at specific (external) hosting platforms [6]. The result is that our control plane manages a potentially large set of DNS name servers that grows and shrinks dynamically over time, which is unlike today’s static and relatively small DNS anycast networks. A precondition is that the control plane is able to interface with the TLD operator’s name servers so that it can send reconfiguration com-
Automatic reconfiguration requires the control plane to collect a rich set of statistics on every DNS anycast node it manages. This includes basic statistics such as processing and storage resources usage, which may be collected using tools such as Nagios. More extended statistics include EDNS Client-Subnet (ECS) extensions (RFC 7871). ECS contains crude geographic information on the location of clients, which the control plane may use to map query demands to the geographical location of end users (i.e., queries’ origin) rather than of resolvers.

Our ultimate goal is that the control plane raises the abstraction level of operating a DNS name server infrastructure, allowing human operators to focus on handling rare incidents because the control plane handles the “regular” ones automatically. We expect this will require advanced visualizations through a TLD operator-wide dashboard that is outside the scope of this article.

We discuss our reconfiguration module and its performance later.

**PRIVACY PROTECTION**

Privacy protection is an important function because the DNS traffic that the control plane analyzes for threat detection and DNS reconfiguration contains IP addresses of resolvers and domain names being looked up, which may constitute Personally Identifiable Information (PII), depending on the jurisdiction.

Figure 2 provides an overview of our control plane, which consists of a high-speed data streaming warehouse called “ENTRADA,” threat detection modules, a DNS reconfiguration module, and a privacy framework.

**ENTRADA**

ENTRADA (Enhanced Top-Level Domain Resilience through Advanced Data Analysis) [7, 9] is our open-source DSW for the TLD control plane. ENTRADA consists of a set of modules that run on top of Apache Hadoop, which is open source as well.

Figure 3 provides an overview of the ENTRADA DSW and how it stores DNS authoritative traffic. Steps I—III refer to domain name resolution. We export the incoming DNS traffic from the .nl authoritative servers to a staging server (step IV), in which the raw PCAP format is converted to an optimized open-source column storage format (Parquet, step V), and later imported into the Hadoop File System (HDFS, VI). Impala provides a massively parallel processing query engine with a standard SQL interface (VII). Applications and services use this interface to connect to ENTRADA.

We choose PCAP as our format for importing DNS traffic from name servers because it includes transport and IP-level headers in addition to their DNS payloads, because it requires few to no changes on name servers (a mirror port on the network or a PCAP process on the name servers), and it is widely used.

ENTRADA delivers the performance we need to build threat detection modules and perform hypothesis tests. For example, we showed in [9] that ENTRADA is able to analyze the equivalent of 52 TB of PCAP data in less than 3.5 min in a four-data-node cluster, using Impala and SQL syntax, which would be infeasible using PCAP format.

Our ENTRADA instance for .nl currently receives DNS traffic from four of our six unicast authoritative name servers and has been operating, and we discuss our privacy protection mechanism below.

**REALIZATION**

Privacy protection is an important function because the DNS traffic that the control plane analyzes for threat detection and DNS reconfiguration contains IP addresses of resolvers and domain names being looked up, which may constitute Personally Identifiable Information (PII), depending on the jurisdiction.

---

2 http://www.nagios.org
3 http://entrada.sidnlabs.nl
4 http://hadoop.apache.org
5 http://impala.io
Figure 3. ENTRADA overview.

Figure 4. Queries to normal and phishing domains.

TDMs dynamically decide which name servers to start or stop at which locations. The DRCM is a logical entity that may be fully distributed across the name servers of a DNS anycast service [6].

Our current DRCM focuses on minimizing the latency between resolvers and the TLD operator’s authoritative name servers. To develop the DRCM, we studied the impact of the number of anycast instances and their physical locations on the latency of the anycast service and reported on this study in [11]. By measuring real-world anycast deployments from C-, F-, K-, and L-Root DNS name servers using the RIPE Atlas framework, we were able to show that a handful of well placed anycast instances provide better and more stable latency than a large-scale infrastructure consisting of several dozens of nodes. For example, C-Root with 8 anycast sites (4 in Europe and 4 in North America) achieved a worldwide median latency of 32 ms, while L-Root with 144 sites (18 x more than C) all over the globe achieved a median latency of 30 ms.

Figure 5 shows the distribution of latency for C-Root and L-Root as seen from around 7900 vantage points around the globe. Note that the larger deployment of L-Root did not result in a shorter distribution tail as well: the 75th percentile of the latency distribution is 76 ms for C-Root and 73 ms for L-Root. These results suggest that connectivity of the anycast site is far more important for the performance of the anycast service than the number of deployed sites, which is an important finding for our DRCM.

We have also set up a worldwide anycast test bed, which we are using to further investigate the relationship between the number of anycast sites and their respective connectivity to service latency, in particular, to understand the efficiency and reliability of anycast deployments.
impact of traffic engineering through anycast for
the mitigation of DDoS attacks. We are actively
probing the anycast infrastructure to understand
the effects of runtime reconfigurations. We also
evaluated the use of the ECS extension based on
real data measured at two name servers that are
authoritative for popular second-level domains
such as apache.com, and we have modified them
to receive and process queries with the ECS
extension.

**Privacy Framework**

Our Privacy Framework protects the privacy of
the users of a TLD [8]. Its key concept is a privacy
policy, which is a structured document in natural
language that defines what data ENTRADA and
its applications process for a particular purpose
using which data filters. A filter is an operation
that ENTRADA or an application applies to per-
sonal data. Examples are pseudonymization and
aggregation. Filters form an essential element in
the Privacy Framework, because they ensure that
the privacy policies are verifiably enforced by
technical means.

ENTRADA application developers and
researchers formally submit privacy policies to
the privacy board, which is a body within the TLD
operator’s organization that reviews policies. The
privacy board approves or rejects the policy and
informs the author through a policy evaluation
report.

After policy approval, the author implements it
as part of a policy enforcement point (PEP), which
is the technical component within ENTRADA or
one of its applications that realizes an approved
privacy policy and actually applies the policy’s
filters at runtime.

Our implementation of the framework for .nl
conforms to both EU and Dutch laws, and we
reported it to the Netherlands Data Protection
Authority. Our privacy board consists of a tech-

cal expert, a legal expert, and a member of our
management team. They approved several priva-

cy policies as of mid-2015, which we activated

**Related Work**

To the best of our knowledge, we are the first to
propose a system that enables TLD operators to
become threat intelligence providers and increase
the robustness of their DNS services. However,
there is scattered prior work on individual compo-


dents.

The operator of the .uk TLD (United King-
dom) developed Turing, a system that appears
to be similar to ENTRADA. Turing, however, is a
commercial closed source solution, and there is
little publicly available information about its tech-

cal implementation. As far as we know, they
did not extend their platform with functions to
dynamically reconfigure name servers; nor did
they include privacy protection mechanisms. We
are also unaware of deployments of Turing at TLD
operators other than at the .uk operator.

There have been several research works that
use DNS TLD data for detection of malicious
domains, but not as part of a larger modular sys-

tem such as our control plane. Hao et al. [12]
analyzed the initial lookup behavior of malicious
domains under .com and .org using a spam trap.

Also, there are different methods to classify mali-
cious websites. For example, Abbasi and Chen
[13] present a comparison of tools to detect fake
websites and perform content analysis to classify
the websites.

The dynamic reconfiguration of DNS anycast
networks is a technique that has been used to
guide clients to the best node of a content distri-
bution network (CDN) in terms of network laten-

cy [6]. But the topic has not been explored before
in the context of TLD operators, which need to
support all networked applications that use the
DNS and cannot assume that the roles of DNS
operator and content provider are collocated as in
the case of [6].

**Conclusions and Future Work**

We present a control plane for operators of
top-level domains in the DNS that enables them
to increase the security and stability of their TLD
by becoming a threat intelligence provider. Our
control plane is a system that extends a TLD oper-
ator’s traditional services and leverages the DNS
traffic and the domain registration transactions
that a TLD operator handles. The control plane
continuously stores and analyzes this data to auto-
matically detect potential threats in the TLD and
shares this information with other players in the
TLD, such as hosting and access providers. It can
also use the information to dynamically scale the
TLD operator’s DNS infrastructure, which increas-
es the robustness of the TLD operator’s DNS ser-


dices.

Our control plane builds on the ENTRADA
open source software, which we have developed
on top of a Hadoop-based data storage cluster.
ENTRADA enables TLD operators to easily feed
their authoritative DNS traffic into the control
plane, to run our threat detection modules, and to
add their own. ENTRADA is currently being used
by at least six operators of country code TLDs.
It comes with a Privacy Framework that enables
TLD operators to manage the personally identifi-
able information of TLD users.

Our future work consists of further refining
and implementing the control plane, for instance,
in terms of modeling the DNS ecosystem using

---

9 http://nominet.uk/turing
and a variety of data sources, extending the control plane to other types of DNS operators, the interfaces a TLD operator needs to provide toward its DNS services, the impact of adding and removing nodes from a DNS anycast network, and new threat detection modules such as for the detection of booter sites.

Acknowledgments

We thank Kees Neggers (SIDN Supervisory Board), Aiko Pras (University of Twente), Marc Groeneweg (SIDN), Moritz Müller (SIDN), and the anonymous reviewers, who provided valuable feedback at various stages of this article.

Ricardo de O. Schmidtm is the work is sponsored by the SAND project (http://www.sand-project.nl).

REFERENCES


Biographies

Cristian Hesselman (cristian.hesselman@sidn.nl) directs SIDN Labs, the research team of the .nl operator, SIDN. His work focuses on mechanisms and systems that further enhance the security, stability, and interoperability of .nl, the Domain Name System, and the wider Internet. He was previously a senior researcher at Telematica Instituut and a software engineer at Lucent Technologies. He holds a Ph.D. and an M.Sc. in computer science, both from the University of Twente, the Netherlands.

Giovane C. M. Moura (giovane.moura@sidn.nl) is a data scientist with SIDN Labs, the research arm of the top-level domain registry of the Netherlands (.nl). He works in research projects that involve performance and security of computer networks. Prior to SIDN, he worked as a postdoctoral researcher at TU Delft, and obtained his Ph.D. from the University of Twente, both in the Netherlands. He has also a Master’s degree in computer science from the Federal University of Rio Grande do Sul, Brazil.

Ricardo de Oliveira Schmidtm (cschmidt@utwente.nl) is a post-doctoral researcher within the Design and Analysis of Communication Systems group at the University of Twente. He obtained his Ph.D. from the same university in 2014. His research interests are in Internet security and management, with research approaches strongly based on Internet measurements and monitoring. He has been involved in various Dutch and European projects, and is currently responsible for running two national projects in management and security of the DNS system.

Kees Toet (kees.toet@sidn.nl) is SIDN’s director of IT, responsible for the operation, security, and stability of the authoritative name servers and the domain registration database of the .nl TLD (The Netherlands). Following his IT education at the Dutch Ministry of Defense, he has worked at several governmental and commercial companies as IT Manager.