COMPUTERS & SECURITY ■■ (2018) ■■–■■

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Toward secure name resolution on the internet

Christian Grothoff [a,*], *Matthias Wachs* [b], *Monika Ermert* [c], *Jacob Appelbaum* [d]

[a] *Bern University of Applied Sciences, Bern, CH, Switzerland*
[b] *TU Munich, Munich, Bavaria, DE, Germany*
[c] *Independent Journalist, Croissant-Rust-Str.15a 81243 Munich, Germany*
[d] *TU Eindhoven, Eindhoven, Netherlands*

ARTICLE INFO

Article history:
Available online

Keywords:
Name resolution
Privacy
Future Internet
Network architecture
Technology and society

ABSTRACT

The Domain Name System (DNS) provides crucial name resolution functions for most Internet services. As a result, DNS traffic provides an important attack vector for mass surveillance, as demonstrated by the QUANTUMDNS and MORECOWBELL programs of the NSA. This article reviews how DNS works and describes security considerations for next generation name resolution systems. We then describe DNS variations and analyze their impact on security and privacy. We also consider Namecoin, the GNU Name System and RAINS, which are more radical re-designs of name systems in that they both radically change the wire protocol and also eliminate the existing global consensus on TLDs provided by ICANN. Finally, we assess how the different systems stack up with respect to the goal of improving security and privacy of name resolution for the future Internet.

1. Introduction

On the net, close to everything starts with a request to the Domain Name System (DNS), a core Internet protocol to allow users to access Internet services by names, such as www.example.com, instead of using numeric IP addresses, like 192.0.2.137 or even worse 2001:DB8:4145::4242. Developed in the "Internet good old times" where privacy and security was not a concern, the contemporary DNS allows DNS operators to monitor user behavior and usage patterns, and exposes information about the existence and availability of most services on the Internet (Bortzmeyer, 2015). Consequently, it attracts all sorts of commercially-motivated surveillance and manipulation: For example, Google's public DNS service permanently logs a dozen items about each request, including the requested domain name.[1] Also, Cisco-owned OpenDNS logs "*any statistical information related to the usage, traffic patterns and behavior of the users*".[2] Finally, there are ISPs manipulating DNS requests and responses, thereby achieving monetary benefits through advertisements.[3] Security problems of these "wildcard" redirections of DNS traffic have been noted, but are ongoing (Why top level domains should not use wildcard resource records, 2015). Furthermore – as new documents of the NSA spy program MORECOWBELL confirm – the National Security Agency as well as other intelligence agencies use the

* *Corresponding author.*
E-mail addresses: christian.grothoff@bfh.ch (C. Grothoff), wachs@net.in.tum.de (M. Wachs), ermert@ermert.info (M. Ermert), j.appelbaum@tue.nl (J. Appelbaum).

[1] See https://developers.google.com/speed/public-dns/privacy.
[2] See https://www.opendns.com/terms-of-service/.
[3] See https://www.wired.com/2008/04/isps-error-page.

https://doi.org/10.1016/j.cose.2018.01.018
0167-4048/

DNS for both surveillance and to drive users to malicious services. This said, DNS remains unprotected against active and passive attacks by malicious entities with much lower capabilities.

DNS currently treats all information in the DNS database as public data. The content of queries and answers is typically not encrypted. This allows passive attackers to monitor the queries of users and see which services they are using and which websites they are visiting. For an on-path active attacker, DNS facilitates locating potentially vulnerable services, which is the first step to their subsequent exploitation with commercially available 0-day attacks.

Given the design weaknesses of DNS, this begs the question if DNS can be secured and saved, or if it has to be replaced – at least for some use cases. The question has also been asked inside the standardization process.[4]

In the last two years, there has been a flurry of activity to address security and privacy in DNS at the Internet Engineering Task Force (IETF), the body that specifies Internet standards, including the DNS. The Internet Architecture Board, the peer body of the IETF, called on the engineers to use encryption everywhere, possibly including DNS (I. A. Board, 2014).

Despite the acknowledgment of the the DNS weaknesses and privacy implications in RFC 7626 (Bhargavan et al., 2015) experts are not expecting that existing industry solutions will change the situation anytime soon:

> "It seems today that the possibility of massive encryption of DNS traffic is very remote." (Bortzmeyer, 2013)

The discussions in the IETF now include proposals for "query name minimization", Confidential DNS, DNS over TLS, DNSCurve and more radical proposals for alternative name system designs to improve privacy. Additional work on encrypting traffic to the authoritative name servers high in the chain is ongoing (Bortzmeyer, 2016a). All of these designs take different approaches in reducing the role of DNS as a source of meta data in the digital panopticon known as the Internet. Before we present the different approaches, we illustrate the security goals and the threat model, using the NSA spy programs as a highly capable attacker and explain what the benefits for the attacker and the risks for the DNS user are. Note that, the NSA is only one of the potential attackers, as other state actors as well as criminals can use the same techniques, and some commercial entities mine data as well to feed their profiling databases. We present the NSA attack as an exemplary, because of their technical capabilities and the explanations of their DNS attack strategies published in recently published documents of the agency itself.

This survey paper makes the following contributions:

- Introduction to the problem of name resolution and the various requirements modern name systems should address.
- Description of recent work on DNS security and privacy by the IETF.
- Survey of alternative secure name systems that have been implemented and aspire to replace DNS.
- Assessment of the advantages and disadvantages of the various approaches, considering their impact on security, privacy, and challenges for deployment.

The remainder of the paper is organized as follows. Section 2 provides an introduction to DNS, establishing terminology that will be used throughout the paper. Section 3 then describes possible security and privacy goals for name systems, and Section 4 provides evidence that nation state adversaries targeting security and privacy vulnerabilities in DNS are active today. Section 5 then describes two types of adversaries that name systems should be concerned about. We then discuss the current trajectory of the evolution of DNS security and privacy under the stewardship of the IETF in Section 6. More radical alternatives for name systems that leave most of the legacy of DNS behind are discussed in Section 7 (Namecoin), Section 8 (GNU Name System), and Section 9 (RAINS). A compact assessment of the different systems is presented in Section 10. Finally, we conclude with the political implications of the state of the art in Section 11.

2. Background: DNS

The Domain Name System (DNS) is an essential part of the Internet as it provides mappings from host names to IP addresses, providing memorable names for users. DNS is hierarchical and stores name-value mappings in so-called *records* in a distributed database. A record consists of a name, type, value and a time-to-live. Names consist of *labels* delimited by dots. The root of the hierarchy is the null label, and the rightmost label in a name is known as the top-level domain (TLD). Names with a common suffix are said to be in the same *domain*. The *record type* specifies what kind of value is associated with a name, and a name can have many records with various types. A well-known record type is the "A" record, which maps names to IPv4 addresses.

The DNS database is partitioned into *zones*. A *zone* is a portion of the namespace where the administrative responsibility belongs to one particular authority. A zone has unrestricted autonomy to manage the records in one or more domains. Very importantly, an authority can delegate responsibility for particular *subdomains* to other authorities. This is achieved with an "NS" record, whose value is the name of a DNS server of the authority for the subdomain. The *root zone* is the zone corresponding to the empty label.

The root zone is operated by the Internet Assigned Numbers Authority (IANA) under the control of a new multistakeholder oversight process in which several stakeholder bodies of ICANN as well as representatives of the IP number registries and representatives from the Internet Engineering Task Force (IETF) have a say. Practically "Public Technical Identifiers" (PTI) is in charge of providing the services. PTI is an affiliate to the Internet Corporation for Assigned Names and Numbers (ICANN).

The root zone contains "NS" records which specify names for the authoritative DNS servers for all TLDs, such as ".de" or ".berlin".

[4] See https://https://tools.ietf.org/html/draft-klensin-dns-function-considerations-04.

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

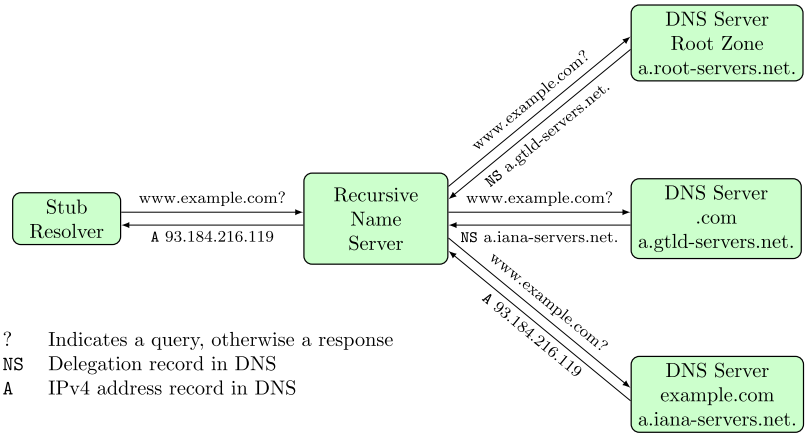


Fig. 1 – Resolving the name www.example.com with DNS. Many operating systems only provide minimal *stub resolvers* forwarding requests to full resolvers. To resolve a name, these resolvers start with querying the name servers of the root zone. If a server cannot provide the required information, it refers the resolver to the next server to query until the server *authoritative* for the respective zone is found.

Names in DNS are resolved using *resolvers*. Many modern operating systems do not provide a full implementation of a DNS resolver but only so-called *stub resolvers*. These stub resolvers do not resolve names directly but forward the request to another resolver. In general, we will refer to resolvers that merely forward requests (and possibly cache replies) as *forward resolvers*. After forwarding, the query eventually reaches a *recursive name server*, which is typically provided by the Internet Service Provider (ISP), as shown in Fig. 1. These recursive name servers resolve the name by first querying the root servers for the required name and by way of recursion go down the DNS tree to fetch the information from the authoritative DNS server. The queried root servers provide the querying resolver with an "NS" record to the server authoritative for the TLD zone, the authoritative server for the zone provides the record for the authoritative server for the domain, subdomain and so on. This *iterative* process is repeated, and terminates for sure when the resolver queries the *authoritative name server* which is responsible for a particular domain.

DNS strongly benefits from caching of DNS information: many *caching resolvers* store information previously requested to improve lookup performance. They use cached record data to skip some or all of the iterations, and thus can return information more quickly to the client.

With the use of forwarding resolvers, the IP address of the client is hidden from authoritative name servers. This gives the user a certain degree of privacy as it prevents operators of authoritative name servers to monitor the source of DNS requests. Naturally, the operators of the forwarding resolvers can still trivially monitor and censor users' requests. Passive dragnet monitoring with systems such as TURMOIL and XKEYSCORE are also able to see any part of the transaction that is available in the ingestion filter.

3. Security goals

When considering improving the security of DNS, there have been striking disagreements among designers as to what the security goals of the DNS system should be. What most designers do agree with is that for the public DNS service, anyone should be able to resolve domain names in it without prior authorization. This does not preclude the possibility of DNS servers returning sensitive records only for certain users, an approach commonly known as *split view*. However, generally speaking, the consensus is that DNS should answer queries without requiring query origin authentication. For this work, we focus on DNS from a security perspective, and thus focus on the following privacy-related security goals:

3.1. *Query origin anonymity*

Even if users of DNS do not have to authenticate, that does not mean that they are anonymous. In the original protocol, the IP addresses of the stub resolvers are hidden behind the recursive name servers, providing a thin veil of privacy. However, this may come at the expense of the origin having to trust the recursive name server. Moreover some ISPs log DNS queries at the resolver to monetize the data. Some even embed user information (e.g. a user id or MAC address) within DNS queries to enable services such as parental filtering or geolocation.

3.2. *Data origin authentication and integrity protection*

Except for regional censors that today block domains by modifying DNS responses, most designers want to see the authenticity and integrity of DNS responses protected. Weak designs simply use secure communication channels between authenticated resolvers. This achieves integrity protection against adversaries in the network but does not help with data authenticity. Another possibility is to cryptographically sign responses with private keys held online; however, as a strong adversary may compromise authoritative name servers, the best protections are achieved by using offline keys for signing zone data to achieve "end-to-end" security including origin authenticity and integrity protection.

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

3.3. *Zone confidentiality*

Before the DNS, all name resolution data was public. With DNS, the notion that zone data could be semi-private and only be exposed upon matching request became a possibility. Exposing full zone information provides useful information to attackers, as they can enumerate network services offered by the target, which with virtual hosting or IPv6 might otherwise not be feasible. Thus, it is desirable to minimize the adversary's ability to enumerate the names in a zone.

3.4. *Query and response privacy*

The DNS query itself or the DNS response may include sensitive information. The design principle of data minimization dictates that participants should only learn as much as necessary, thus some proposals try to make DNS less chatty. In the most extreme case, a domain name may contain a password, and responses might contain key material, which both ought to be kept confidential from the recursive and (online) authoritative name servers.

3.5. *Censorship resistance*

A special goal of some name systems is resistance against censorship. The goal is to make it impossible even for governments that have jurisdiction over any possible DNS operator to block name resolution using legal attacks. This is typically achieved by designs that are self-organizing and thus do not require the interaction with TLS registries.

3.6. *Traffic amplification*

While censorship resistance covers availability of the name system itself, DNS is also frequently used as an attack vector against networks. By sending relatively small DNS queries to (open) DNS resolvers from spoofed origin addresses, DNS can be used to amplify the network bandwidth available to an adversary (Paxson, 2001).

3.7. *Application-level timing attacks*

Name service responses are often cached. This gives rise to timing attacks against caches, especially if an application uses a cache and provides untrusted code the opportunity to probe the cache. This is particularly prevalent with Web browers which both cache DNS replies and may allow untrusted Web sites to probe the cache, thereby possibly disclosing which names were recently resolved by the user (Felten and Schneider, 2000; Krishnan and Monrose, 2010). Naturally, cache snooping is not limited to application-level caches, but also applies to caches on the network.

4. Exemplary attacker: The NSA's MORECOWBELL and QUANTUMDNS programs

Threats against the DNS and its users are not theoretical. As set of top secret documents published by Le Monde (Eudes et al., 2015) revealed, the American spy agency NSA monitors DNS as a source of information about the Internet (Fig. 2a). NSA's MORECOWBELL program uses a dedicated covert monitoring infrastructure to actively query DNS servers and perform HTTP requests to obtain meta information about services and to check their availability (Fig. 3).

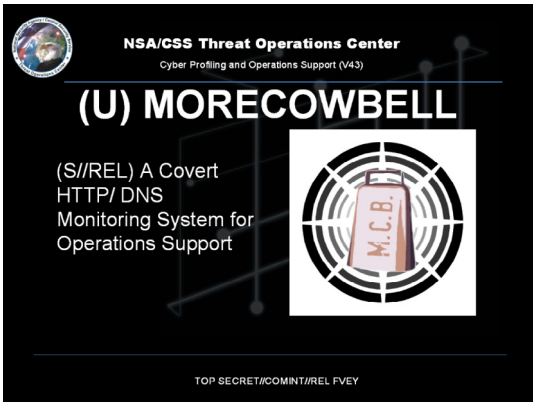
Despite the open nature of DNS, the NSA does so covertly (Fig. 2b) to ensure the thousands of DNS lookups every hour are not attributed to the US government (USG). In fact, the servers the NSA rented for the purpose of monitoring DNS and checking Web servers using HTTP are located in Malaysia, Germany and Denmark (Fig. 2c), allowing the NSA to perform the monitoring covertly and to get a more global view on DNS name resolution and service availability. While the NSA slides only list these three countries, the PACKAGEDGOODS non-attributable monitoring infrastructure that MORECOWBELL builds on is known to span machines in at least 13 other countries, as described previously by Der Spiegel in a set of slides describing the NSA's TREASUREMAP program (N. T. O. C. (NTOC), 2014).

What is interesting is that at the time, the NSA did not care much about the specific content of the Web servers or the DNS entries – as usual the NSA is after the meta data: the NSA wants to know if the DNS information has changed, and check on the availability of the service. The slides show that this simple check has some rather benign uses, for example it is used to monitor some of the US government's own websites.

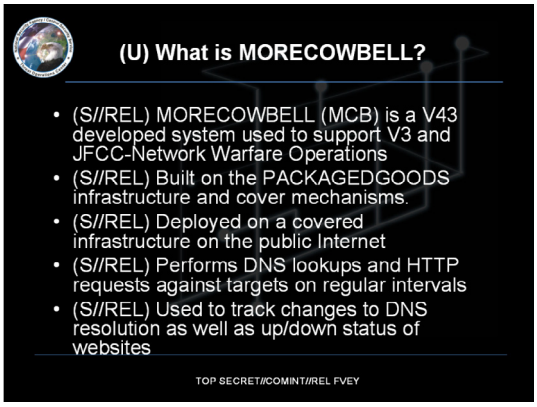
A key justification for the need to make the active probing of DNS unattributable to the US government is most likely its use for "Battle Damage Indication" (Fig. 2d). Specifically, after "Computer Network Attacks (CNA)" are used against critical network infrastructure, the US may use such probes to confirm that its attacks have found their targets when the lights go out on the Internet systems, say of some foreign government. By monitoring for changes in the DNS, the attack could be repeated if the victim tries to shift its services to another system or network. By keeping the monitoring infrastructure covert and geographically distributed, the NSA gets a global view on the impact of an attack. This makes it harder for victims to identify the monitoring servers, which otherwise might enable victims to evade the attack by treating requests from monitors differently.

The various documents of the NSA relating to DNS show that existing covert attacks on DNS enable mass surveillance and active attacks (Weaver). With the revelation about the NSA's QUANTUMTHEORY family of projects (Fig. 2e) with subprojects like QUANTUMDNS (Fig. 2f), we know that powerful attackers like nation states can not only eavesdrop DNS traffic but also inject DNS responses to modify the result of name resolution or make it even completely fail (Redacted (NSA, S32X), 2014). With DNS not providing confidentiality to protect a user's privacy, it is easy to create a profile of the users and their surfing behavior on the Web (Krishnan and Monrose, 2010). This information could then also be used to perform QUANTUMTHEORY attacks against the target. NSA programs like QUANTUMBOT have the purpose to monitor IRC botnets and detect computers operating as bots for a botnet and hijack the command and control channel to manage the bots. Note that the goal here is not necessarily to disable the bots, but to control them. These programs using DNS as a first stage of

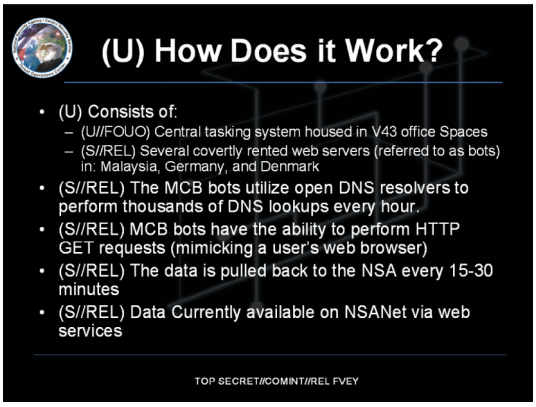
Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018



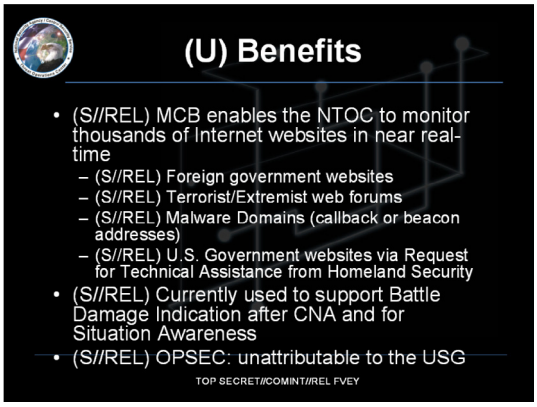
(a) A Covert HTTP/DNS Monitoring System



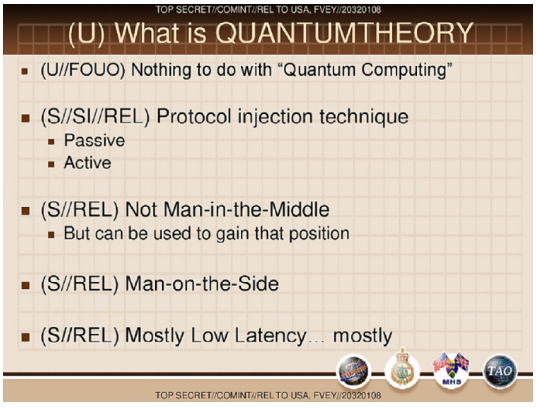
(b) What is MORECOWBELL.



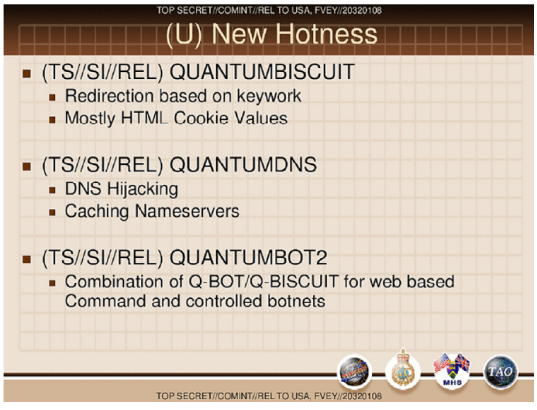
(c) How does MORECOWBELL work?



(d) "Benefits" of MORECOWBELL.



(e) QUANTUMTHEORY: Man-on-the-side attack.



(f) QUANTUMDNS.

Fig. 2 – Slides about MORECOWBELL and QUANTUMDNS. Attacks on DNS are not theoretical. Other slides from the NSA say that QUANTUMDNS is operational and has been successfully used.

network-based attacks are evaluated by the NSA to be *highly successful* according to their documents (A. (NSA), 2014).

Thus, the Internet community needs to work toward technical solutions resolving the privacy and security issues with name resolution and the current Domain Name System (DNS), especially given that legislative initiatives to protect citizens generally contain loopholes exploited by spy agencies as "the law maintains an old-fashioned focus on physical materiality" (Arnbak and Goldberg, 2015). In the next step, we will review a range of current proposals that have been made to improve the security of this critical Internet service.

5. Adversary model

To evaluate existing approaches aiming to improve name resolution security and privacy, we employ two different adversaries:

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

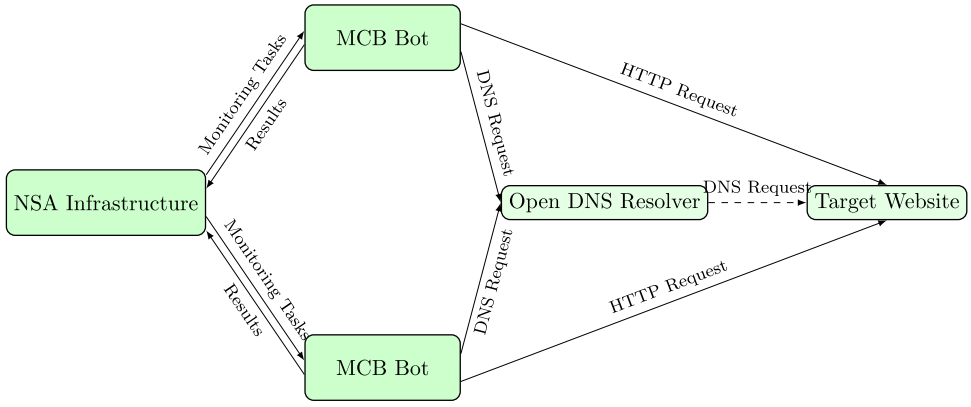


Fig. 3 – From Eudes et al. (2015): NSA's MORECOWBELL infrastructure: a list of targets to monitor is deployed to geographically distributed bots performing DNS and HTTP requests against target websites to collect information about the availability of services. The resulting data are returned to the NSA in regular intervals.

On the one hand, we examine adversaries within the name system. This can be DNS infrastructure providers operating DNS relevant systems including DNS recursive or forward resolvers. Such adversaries can be honest-but-curious interested in users' usage patterns by monitoring name resolution. This is a particularly common adversary, as often operators consider this behavior as perfectly legitimate monitoring of their own infrastructure. However, just like the mail service does not have a right to read one's mail, an ISP does not have the right to inspect the personal information of its users, even if it traverses its systems. To counteract such an adversary query origin anonymity and query response privacy are relevant security goals. Besides being curious, such an adversary may be interested in modifying results or make name resolution fail, requiring integrity protection, data origin authentication, and censorship resistance as security goals to antagonize such an attacker.

On the other hand, we employ very powerful adversaries as introduced with the NSA and its MORECOWBELL and QUANTUMDNS programs. Such adversaries may be interested in monitoring users' behavior and monitoring DNS resolution by being able to eavesdrop network traffic, requiring query origin anonymity and query response privacy as a countermeasure. Besides monitoring, such adversaries may want to tamper with name resolution by modifying name resolution (requiring integrity protection and data origin authentication as security goals) or make name resolution fail using technical or legal means (requiring censorship resistance for name systems). Such adversaries may exploit name systems by obtaining zone information to learn about network services that they may subsequently target and exploit. Here, zone confidentiality and response confidentiality are important to avoid leaking knowledge about potential targets.

6. Evolution of DNS

This section discusses various evolutions of the DNS protocol which are largely driven by the IETF with backwards-compatibility in mind.

6.1. DNSSEC

The Domain Name System Security Extensions (DNSSEC) (Arends et al., 2005) add integrity protection, data origin authentication for DNS records and secure denial of existence. DNSSEC does not attempt to improve privacy. It adds record types for public keys ("DNSKEY"), delegation signer ("DS"), for signatures on resource records ("RRSIG") and secure denial of existence ("NSEC"). Fig. 4 illustrates the interactions among resolvers using DNSSEC. DNSSEC creates a hierarchical public key infrastructure in which all DNSSEC zone operators must participate. It establishes a trust chain from a zone's authoritative server to the trust anchor, which is associated with the root zone. This association is achieved by distributing the root zone's public key out-of-band with, for example, operating systems. The trust chains established by DNSSEC mirror the zone delegations of DNS.

Currently DNSSEC uses primarily the RSA crypto system (the root zone has used a set of RSA-2048 and RSA-1024, a switch to RSA-2048 is prepared)[5], which must be supported by every DNSSEC-enabled resolver. The IETF has started to add additional ciphers based on elliptic curves (Hoffman and Wijngaards, 2012). Regardless, DNSSEC generally requires longer DNS responses to transmit the public keys and signatures. To do this, DNSSEC relies on DNS extensions (EDNS0) which increases the UDP packet size limit from 512 to 4096 bytes.

Like DNS, DNSSEC allows for negative replies (NXDOMAIN). To enable secure claims of non-existence, DNSSEC needed a way to create a signed statement that records do not exist. As DNSSEC was designed to keep the signing key offline, "NSEC" records were introduced to certify that an entire range of names was not in use.

6.1.1. Analysis

The use of RSA in DNSSEC leads to unnecessarily large keys and signatures, and the effect is amplified because response

[5] https://www.icann.org/resources/pages/ksk-rollover, accessed 2017, see also Ermert (2016).

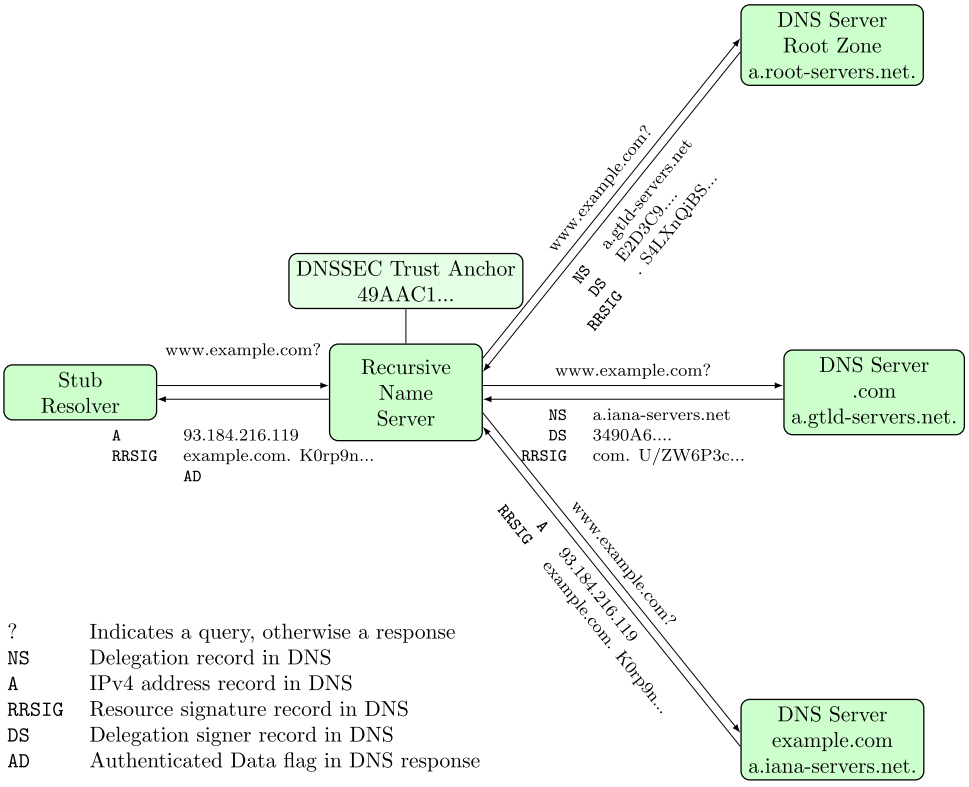


Fig. 4 – Resolving the name www.example.com with DNS and DNSSEC: information returned by name servers is cryptographically signed to ensure authenticity and integrity. This information is stored in "RRSIG" records and information about the parent zone stored in "DS" records. A resolver can verify a signature by following this trust chain and using the *trust anchor* shipped out-of-band. Stub resolvers cannot verify this chain and the resolver therefore indicates to the stub resolver that it checked authenticity by setting the AD bit in the response given to the client.

includes the signatures for all of the signature schemes supported by the authoritative server. This can result in message sizes that exceed traditional size restrictions on DNS packets, possibly leading to additional vulnerabilities (Herzberg and Shulman, 2013). The introduction of additional cipher suites is not an optimal solution, as deploying multiple ciphers either further increases packet size and computational cost (if both ciphers are used to secure the same delegation), or reduces security to the weaker of the two ciphers if a mixture of ciphers is used on the resolution path.

The increase in the message size also increases the risk of traffic amplification created by DNS by allowing an adversary to achieve a larger traffic amplification factor (van Rijswijk-Deij et al., 2014). Consequently, DNSSEC ranks highly in terms of amplification potential considering the combination of the number of available servers for amplification and the achievable amplification multiplier (Rossow, 2014).

DNSSEC effectively lifts the few traditional limitations on bulk acquisition of zone data, practically reducing zone confidentiality. Before DNSSEC, DNS zone administrators could disallow zone transfers, making it difficult for an adversary to systematically enumerate all of the DNS records in a zone. With NSEC records, looking at the boundaries of those ranges allows an adversary to quickly enumerate all names in a zone that are in use. An attempt to fix this via the introduction of "NSEC3" records has been described as broken by security researchers[6]. Nevertheless, NSEC3 is now widely used.[7] As a result, DNSSEC makes it even easier for an adversary to discover vulnerable services and systems (Bau and Mitchell, 2010). The introduction of NSEC5 records (Goldberg et al., 2014) may finally restore zone confidentiality.

Finally, TLD operators, especially for ccTLDs, are often subject to the same jurisdiction as the users and service operators that primarily use the zone. As a result, legal proceedings to censor a service are generally quite effective against these trust chains. Thus, DNSSEC does not significantly improve censorship resistance.

6.2. *Query name minimization*

The recent discussions in the IETF to improve privacy in DNS (in a dedicated DPrive Working Group) include a standard for so-called *query name minimization* or *QNAME minimization* (Bortzmeyer, 2016b). Query name minimization slightly improves query privacy by having recursive name servers not send the full query to the DNS servers contacted in each resolution step. Instead, each DNS server only receives as much of

[6] https://dnscurve.org/espionage2.html.
[7] http://secspider.verisignlabs.com/stats.html.

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

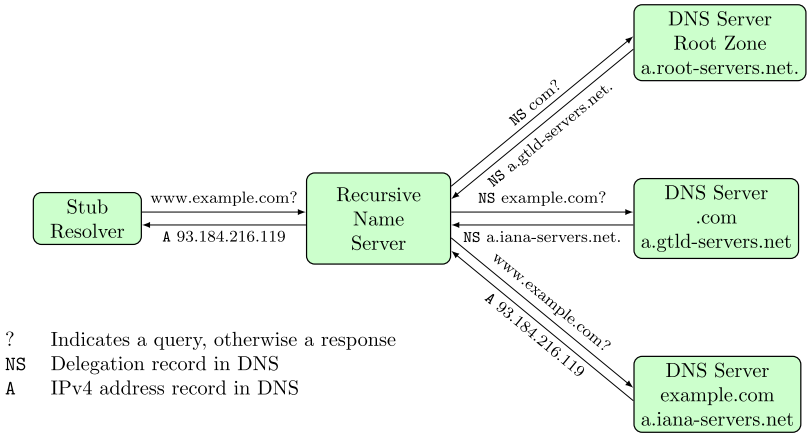


Fig. 5 – With query name minimization, resolving the name www.example.com no longer exposes the full name and query type to the root zone and the .com authority. Naturally, this scheme still leaks quite a bit of sensitive information to the TLD's DNS server. Furthermore, the effect is even weaker in practice, as root zone is already often not contacted as information about TLD name servers is typically cached at forwarding resolvers.

the DNS name as is necessary for making progress in the resolution process (Fig. 5). Consequently, the full name being queried is typically only exposed to the final authoritative DNS server.

6.2.1. *Analysis*

Query name minimization can simply be implemented by changing how recursive name servers construct their iterative queries. Query name minimization may negatively impact performance, as at least in theory the full query may enable the DNS servers to respond faster with the ultimate answer, if cached information is available or they are the authoritative server for the queried fully qualified domain name. Even with query name minimization, the recursive name servers (at an ISP for example) still learn the full query and reply of a user.

Query name minimization has the advantage that its deployment only requires changes to the recursive name server, and the disadvantage that the change is entirely outside of user control.

Query name minimization can be combined with the various approaches to encrypt DNS traffic presented in the next sections. Without query name minimization, simply encrypting DNS traffic – for example using TLS as described in the following section – continues to expose the full query to many DNS servers, in particular root servers and authoritative servers for the respective TLD.

6.3. *T-DNS: DNS over TLS*

DNS over TLS (standardized as RFC 7858 (Hu et al., 2016)) simply transmits DNS queries and responses over the well-known Transport Layer Security (TLS) protocol. The performance loss associated with this change is mitigated by re-using a TCP connection for multiple DNS requests with moderate timeouts, pipelining requests and allowing out of order processing. This way, the DNS over TLS promises reasonable performance despite the overheads from TCP and TLS.

DNS over TLS is available as part of the Unbound, Knot and Bind DNS servers. Several pilot public servers implementing DNS over TLS have been set up. The Quad9 public DNS server network also offers TLS for queries and answers.

6.3.1. *Analysis*

By switching to TCP from connectionless UDP, DNS over TLS eliminates the abuse of DNS for traffic amplification (Mankin et al., 2014). However, the additional cost of managing TLS sessions by the DNS servers may increase the susceptibility of DNS itself to denial of service attacks.

Using DNS over TLS does not improve query origin anonymity since the protocol still leaks meta data, allowing third parties to easily determine which DNS data a user accesses. In DNS over TLS, TLS is used in combination with the traditional DNS lookup paths, which may involve the use of forward resolvers that assist endpoints performing DNS queries. The involvement of such forward resolvers can obscure the user's IP address from the other DNS servers; naturally, for this to be sufficient the forward resolvers themselves would have to be trusted to not spy on the user.

Finally, TLS itself does not have the best security track record, with dozens of issues in recent years ranging from high-profile certificate authority compromises to broken implementations and insecure cipher modes (Holz, 2013).

6.4. *DNSCurve*

The first practical system that improves confidentiality with respect to DNS queries and responses was DNSCurve (Bernstein, 2008). In DNSCurve, session keys are exchanged using Curve25519 (Bernstein, 2006) and then used to provide authentication and encryption between caches and servers. DNSCurve improves DNS with respect to query and response confidentiality and hop-by-hop integrity without the need to create expensive signatures or (D)TLS sessions. Specifically, DNSCurve achieves the same round trip time (RTT) as DNS by embedding the public key of the server in the "NS" record, conflating the DNS namespace with key information.

DNSCurve creates an authenticated and encrypted association between a *DNSCurve server* and a *DNSCurve cache*, the

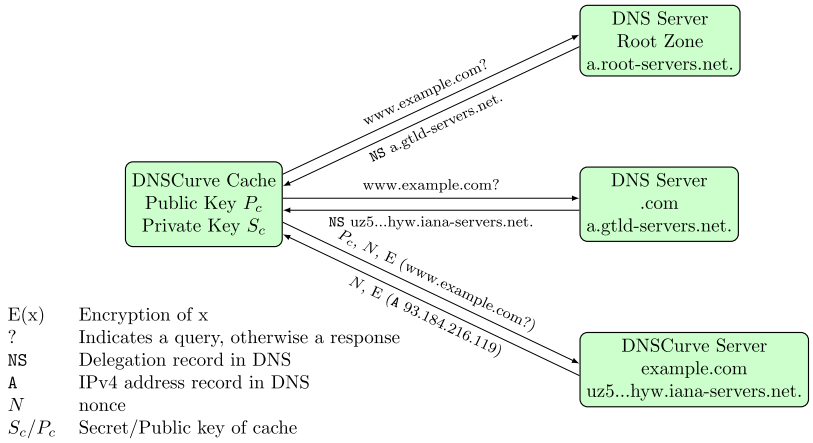


Fig. 6 – Resolving the name www.example.com with DNSCurve. With DNSCurve, the resolving cache and the DNSCurve server exchange a shared secret to encrypt their communication. The DNSCurve server's public key is encoded in the name of the name server itself using Base32. When a DNSCurve cache resolves a name and finds the name server to support DNSCurve, the cache creates a shared secret based on the server's public key, the cache's private key, and a one-time nonce. The cache sends its public key, the nonce and the query encrypted with the shared secret. The server will respond with the result of the query encrypted with the shared secret. The first two lookups to the root zone and the ".com" TLD do not use DNSCurve in the illustration as those currently do not support DNSCurve.

latter being a caching recursive DNS resolver running at the endpoint instead of a DNS stub resolver (Fig. 6). As DNSCurve does not use signatures, the DNSCurve cache cannot prove the authenticity of the cached records to other users, limiting the utility of each cache to the respective endpoint.

6.4.1. *Analysis*

While in DNSCurve the user no longer has to trust a forward resolver, the endpoint's IP address is now directly exposed to the authoritative DNS servers: it is no longer obscured by recursive name servers operated by network service providers. Thus, DNSCurve can increase privacy against an adversary monitoring DNS traffic on intermediary systems or with other cable tapping, but reduces query origin anonymity with respect to authoritative DNS servers, as they learn both the full query and the identity (IP address) of the user. Another commonly voiced concern about DNSCurve is the need to keep private keys online. DNSCurve also cannot protect against censorship, as certain governments continue to effectively control the hierarchy of registrars and can thus make domains disappear. With respect to attacks from the NSA, DNSCurve only helps users against passive surveillance on the wire by protecting the confidentiality of at least the DNS payload.

With DNSCurve, authoritative DNS servers remain a juicy target for mass surveillance. Furthermore, as with DNS, the well-known and easily located DNS servers remain a target and confirmation vector for attacks on critical infrastructure. With DNSCurve, the need for online public key cryptography by the DNS authorities may open up an additional vulnerability to computational denial of service attacks if a small CPU is used to handle a high-speed link.

DNSCurve, like DNS, can be used for traffic amplification. However, due to the use of Curve25519, requests and responses both grow modestly. Thus, the traffic amplification potential of DNSCurve is smaller than that of DNS or DNSSEC.[8]

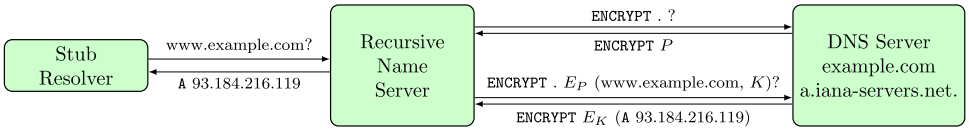
6.4.2. *DNSCrypt*

DNSCrypt is an unstandardized but documented protocol largely based on DNSCurve. It protects the end user's stub resolver queries from network surveillance and tampering thereby improving query and response privacy and integrity. As it is based on DNSCurve, it does not solve any of the major other privacy or security issues present in DNS. The largest known resolver to support DNSCrypt is OpenDNS. There are a number of open DNSCrypt resolvers run by the DNSCrypt community. Today, DNSCrypt remains the most widely deployed DNS encryption protocol designed to prevent surveillance of end users from the network. However, it only helps to solve half of the privacy problem, and it is not widely adopted or standardized.

6.5. *Confidential DNS*

Another IETF draft which has been discussed in the IETF DPrive Working Group suggests an alternative method for adding encryption to DNS. It uses the main extension mechanism of DNS, the introduction of additional record types, to encrypt DNS traffic (Wijngaards, 2014), thereby achieving query and response privacy and integrity protection. With Confidential DNS (Fig. 7), a new "ENCRYPT" record type is introduced to provide the necessary public key that would allow the recursive name server to encrypt the connection to the DNS server. This "ENCRYPT" record contains the public key of the DNS server to be used to encrypt communication initiated by the resolver. The "hack" used by DNSCurve where the public key was added into the "NS" response of the delegating zone is avoided.

[8] http://dnscurve.org/amplification.html.



?	Indicates a query, otherwise a response
.	Query for the root zone
P	Public key of server
K	Encryption Key
E_P (x)	Encryption of x with P
E_K (x)	Encryption of x with K
A	IPv4 address record in DNS
ENCRYPT	Encrypt record in DNS

Fig. 7 – Resolving the name www.example.com with opportunistic Confidential DNS. The resolver retrieves the DNS servers public key querying for the new "ENCRYPT" record. This public key can then be used to encrypt the query to the server. The resolver sends the query encrypted with the server's public key containing the query and the key to encrypt the reply with.

The current draft supports two different operation modes: an *opportunistic* mode which is easier to realize since it does not require major changes to DNS infrastructure and an *authenticated* mode, where a domain's public keys are also stored in the respective parent zone, thus requiring support from the parent zone's DNS infrastructure.

With the opportunistic mode, the public key is no longer associated with the parent zone and instead served separately in the clear and, unless used in combination with DNSSEC, without authentication as a record with the target zone. As a result, Confidential DNS using the "ENCRYPT" record may result in so-called *opportunistic encryption*, which is encryption that is trivially bypassed by a man-in-the-middle attack, as it uses unauthenticated keys for encryption. Storing the encryption key in the zone also implies that encryption requires an extra round of communication to look up the key.

The draft on Confidential DNS provides also a method to achieve authenticated encryption without an extra round trip by storing a domain's public key in the respective parent zone. To do so, Confidential DNS extends DNSSEC's Delegation Signer ("DS") resource records to provide the encryption key for the zone. This resembles the "NS" record used by DNSCurve. This approach makes Confidential DNS susceptible to censorship attacks since it relies on DNS's hierarchical architecture.

The use of a new record type also creates the opportunity for the necessary complexity of a committee-engineered solution: Confidential DNS can use symmetric or asymmetric cryptography, and sports support for 512-bit RSA and AES in CBC mode (which was recently used to finally kill off SSL3 (Möller et al., 2014)).

The draft provides for a variety of failure modes, such as "fallback to insecure" allowing clients to relapse to insecure modes with "leaps of faith" even after secure connections used to be available. Confidential DNS allows implementations to "fallback to insecure" in case one side uses cryptographic algorithms that the other does not support. These various scenarios in which Confidential DNS simply falls back to unencrypted channels (without any indication to the user) highlight how much the design focuses on being easy to deploy at the expense of providing predictable security.

6.5.1. *Analysis*

Overall, the draft fails to set a strong minimum baseline, making it impossible for the system to provide clear assurances. This lack of well-defined security semantics is deadly, as applications cannot rely on the confidentiality of Confidential DNS, even if it were universally deployed.

With the adoption of DNS-over-TLS the draft recently has not been further updated and remains unfinished.

7. Namecoin

Alternative peer-to-peer name systems provide more radical solutions to secure name resolution. Timeline-based systems in the style of Bitcoin (Nakamoto, 2008) have been proposed to create a global, secure and memorable name system (Swartz, 2011). Here, the idea is to create a single, globally accessible timeline of name registrations that is append-only. Timeline-based systems rely on a peer-to-peer network to manage updates and store the timeline. In the Namecoin system (Kraft, 2017), modifications to key-value mappings are attached to transactions which are committed to the timeline by mining. Mining is the use of brute-force methods to find (partial) hash collisions with a state summary (fingerprint) representing the complete global state – including the full history – of the timeline.

Given two timelines with possibly conflicting mappings, the network accepts the timeline with the longest chain as valid, as it represents the largest expense of computational power. This is supposed to make it computationally infeasible for an adversary to produce an alternative valid timeline. This assumes limited computational power and may not actually be binding for certain adversaries.

To perform a lookup for a name with Namecoin, the client has to check the timeline if it contains an entry for the desired name and check the timeline for correctness to ensure that the timeline is valid. To do so, the user has to possess a full copy of the timeline (Fig. 8), which had a size of about 4.7 GB

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

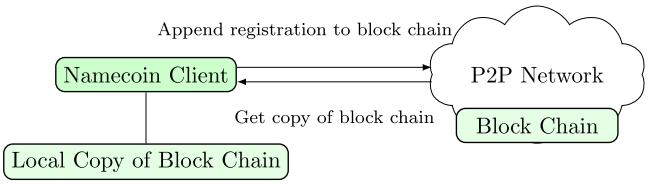


Fig. 8 – The Namecoin name system is decentralized and uses a peer-to-peer network. To achieve a consensus about names registered, Namecoin uses a *block chain* stored in the peer-to-peer network. To register a name, clients have to perform some computational work to get their name appended to the chain. To resolve a name, clients have to possess a full copy of the block chain and search for the name to resolve in the block chain.

in November 2016.[9] Alternatively, users may use a trusted name server participating in the Namecoin network.

7.1. *Analysis*

Unlike the different variations on DNS, Namecoin is designed to withstand legal attacks. Depending on their reach, governments, corporations and their lobbies can legally compel operators of DNS authorities to manipulate entries and certify the changes. Hence, DNS-based systems are vulnerable to censorship at the authoritative servers. With a solution using block chains, regulation is significantly more difficult, as (unprovisioned) block chains do not have a legal entity associated with them.

Namecoin can improve user privacy if the full block chain is replicated at the user's end system. In this case, resolving a name does not involve the lookup and is thus perfectly private with respect to query origin anonymity and query and response privacy. However, replicating the full block chain at each user may be impractical for some devices should Namecoin ever grow to be a serious competitor for DNS. Namecoin also does not protect the zone information from monitoring, and in particular zone enumeration is trivial. However, the decentralized nature of Namecoin does ensure that at least battle damage indication against a name server no longer makes sense.

8. The GNU name system

The authors of this article are working on the GNU Name System (GNS) (Wachs et al., 2014), which is a more radical proposal to address DNS privacy and security issues, and which like Namecoin significantly departs from DNS's name resolution process. The GNS resolution process does not use resolvers querying DNS authorities. Instead, GNS uses a peer-to-peer network and a distributed hash table (DHT) to enable resolvers to look up key-value mappings. However, GNS does not simply replicate information from DNS into a DHT–like approaches focusing on resilience and performance such as CoDoDNS (Ramasubramanian and Sirer, 2004) or DDNS (Cox et al., 2002) – but instead provides a fully decentralized name system conceptually independent from DNS.

GNS is privacy-preserving since queries and responses are encrypted such that even an active and participating adversary can at best perform a confirmation attack, and otherwise only learn the expiration time of a response. Note that the queries and responses themselves are encrypted, not the connections between a resolver and some authority. As all replies are not just encrypted but also cryptographically signed, GNS provides integrity protection since peers in the DHT cannot tamper with the results without immediate detection and data origin authentication.

Due to the use of a DHT, GNS avoids DNS complications such as glue records and out-of-bailiwick lookups. In GNS, the labels of a name correspond precisely to the lookup sequence, making the complete trust path obvious to the user. Finally, the use of a DHT to distribute records also makes it possible for GNS authorities to operate zones without visible, attributable critical infrastructure that could be used for battle damage indication.

GNS can securely resolve names to any kind of cryptographic token. Thus, it can be used for addressing, identity management and as an alternative for today's battered public key infrastructures.

GNS records are about as compact as those of DNSCurve. However, there is no real potential for traffic amplification as the DHT relies on secure connections with a proper handshake that prevents spoofing attacks. The DHT connections are long-lived, limiting the performance impact of the cryptographic handshake compared to DNS over TLS.

8.1. *Names, zones and delegations*

A GNS zone is a public–private key pair and a set of associated records. The GNS name resolution process basically resolves a chain of public keys. GNS uses the pseudo-TLD ".gnu" to refer to the user's own zone, which is called the *master zone*. The master zone provides an alternative to hierarchical addressing, and allows GNS to operate even in the absence of a globally recognized and operational *root zone*. Each user can create any number of zones, but one must be designated as the master zone. Users can freely manage mappings for the labels in their zones. Most importantly, they can delegate control over a subdomain to any other zone (including those operated by other users) using a "PKEY" record, which simply specifies the public key of the target zone. "PKEY" records are used to establish the aforementioned delegation path. Due to the use of a DHT, it is not necessary to specify the address of some system that is responsible for operating the target zone. Record validity in the DHT is established using signatures and controlled using expiration values.

8.2. *Cryptography for privacy*

To enable other users to look up records of a zone, all records for a given label are stored in a cryptographically signed block in the DHT. To maximize user privacy when using the DHT to look up records, both queries and replies are encrypted and

[9] https://bitinfocharts.com/namecoin/.

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

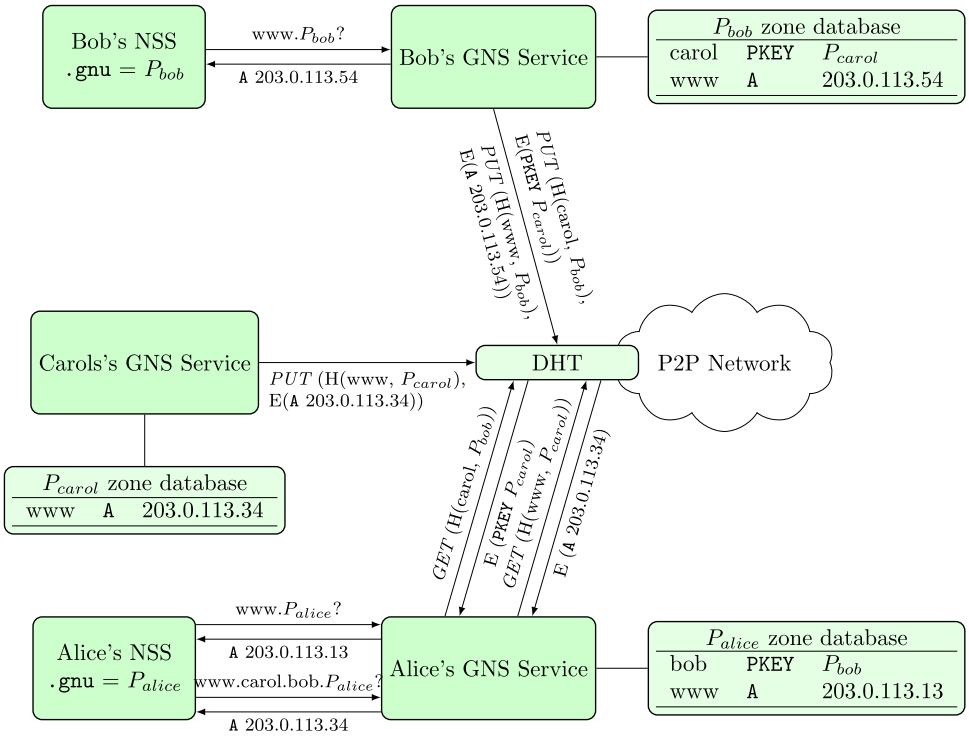


Fig. 9 – The GNU name system: with GNS, every user maintains their own databases containing record sets under labels organized in zones. A zone is referenced by a public-key pair. Here Alice, Bob and Carol have Web servers all reachable under www.gnu. For Alice www.gnu resolves to a different address than for Bob or Carol, as their respective local name service switches (NSS) associate a user-specific public key with .gnu. To allow other users to resolve the names, a user's public zone information is encrypted and published in a DHT under an obfuscated query key. A user can *delegate* to another user's namespace from his local namespace to resolve foreign names. Alice can access Bob's namespace by delegating control over the name bob to P_{bob} in her namespace using a GNS-specific "PKEY" record. This way Alice can access Carols's Web server using the name www.carol.bob.gnu.

replies are signed using a public key derived from the public key of the zone and the label (Fig. 9). Any peer can easily validate the signature but not decrypt the reply without prior knowledge of the public key and label of the zone. Consequently, users can use passwords for labels or use public keys that are not publicly known to effectively restrict access to zone information to authorized parties.

8.3. *Analysis*

Due to the use of a DHT, all GNS queries go to the same fully decentralized and shared global infrastructure instead of operator-specific servers. This provides censorship-resistance and makes it impossible to target a zone-specific server because all machines in the DHT are jointly responsible for all zones – in fact, the key-value pairs do not reveal which zone they belong to. At the same time, encryption and authentication of the records is critical as it helps protect the users from effective censorship or surveillance.

However, unlike the other less radical proposals to overhaul DNS, deploying GNS will be a significant challenge: GNS requires more significant changes to software, as well as a community effort to operate a DHT as a new public infrastructure.

9. RAINS

RAINS is a replacement for DNS for SCION, which itself is a clean-slate Internet architecture. In SCION, Internet autonomous systems are organized into so-called isolation domains (ISDs), which are trust domains isolated against (external) misconfiguration or routing failure. RAINS substitutes having a global DNSSEC trust anchor with a trust anchor per isolation domain. As everybody in an isolation domain relies on the trusted computing base (TCB) of the isolation domain for routing, it is natural for them to also rely on the same TCB for naming. RAINS's authors consider protection against zone enumeration a non-goal (Perrig et al., 2017).

While RAINS is implemented as a complete rewrite of the protocol, the high-level result is largely a combination of known techniques. RAINS combines the resolver structure of DNS, the signed records of DNSSEC, the use of TLS for authenticated encryption for queries and responses (similar to DNS-over-TLS), and incorporates certificates for SCION's host authentication (similar to TLSA records in DNSSEC). The architecture assumes that the IDS's key material and a pinned TLS certificate of the local RAINS resolver are available to the client. Fig. 10 illustrates the lookup

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

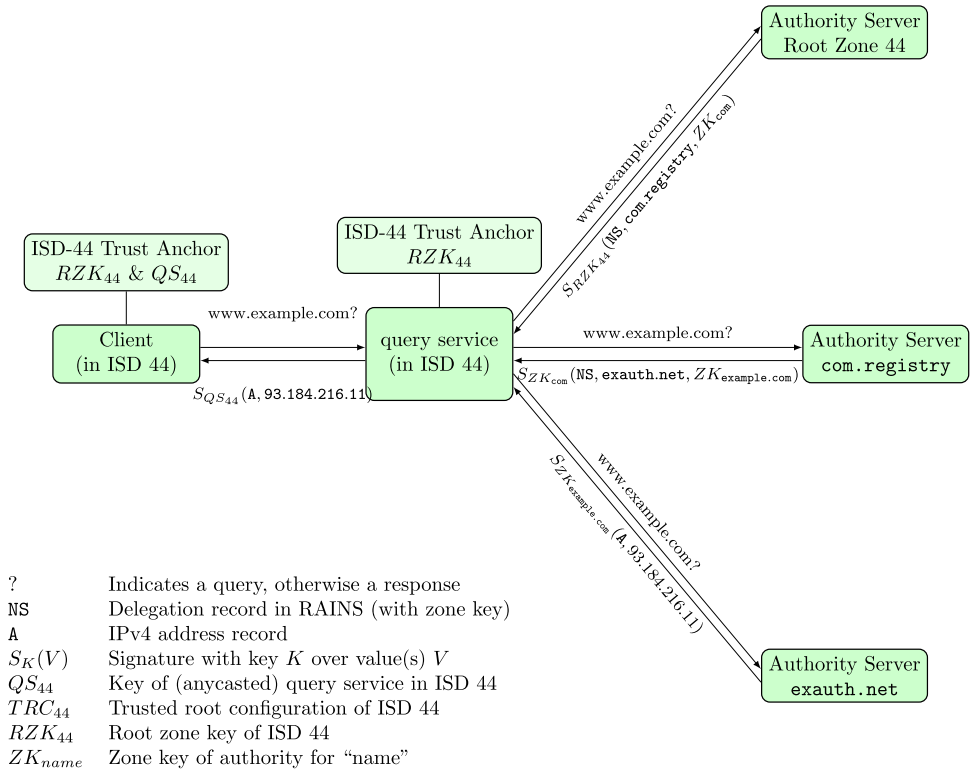


Fig. 10 – Name resolution in RAINS proceeds along similar lines as in DNS, except network links use TLS (with pinned certificates used to bootstrap), DNSSEC-style record signing is mandatory. The global DNSSEC trust anchor is replaced by the trusted computing base of the respective isolation service domain (ISD), in this example TRC_{44} for ISD 44.

procedure, which by design resembles Fig. 4, except without much backwards-compatibility to DNS and without a global trust anchor.

In RAINS, it is possible that the recursive resolver simply signs that it verified the result (as shown in Fig. 10 with $S_{QS_{44}}$). Alternatively, the client may request the complete proof and verify it itself against the root zone key of its isolation domain.

9.1. *Analysis*

Like GNS and DNS over TLS, RAINS does not introduce traffic amplification vulnerabilities. Unlike GNS, RAINS does not attempt to hide the contents of queries or replies from resolvers in the network.

The main difference to DNSSEC over TLS is in the use of the isolation domain as the trust anchor for the root zone. As a result, it is possible for different isolation domains to certify different values for the same name, which conflicts with the goal of having a globally consistent namespace. RAINS addresses this via a naming consistency observer (NCO): name assertions made in one ISD can be observed in any remote ISD. The NCO is a service which continuously monitors name spaces for inconsistencies, with the goal of making illegitimate behavior public. Naturally, such monitoring is inconsistent with protections against zone enumeration.

10. Assessment

The technical approaches presented differ widely in their security goals. We summarize the key differences in Table 1.

Traditional DNS basically assumes a trustworthy IP network, the other models assume that the network cannot be fully trusted to protect the integrity of the data. Protecting the integrity of the responses has thus been the first order of business for all approaches to secure DNS, starting with DNSSEC.

DNSSEC's limited focus means that it does not consider privacy implications of exposing requests and responses and their origin to the network. Only NameCoin and GNS try to hide the nature of client requests from the operators of the network. Here, GNS is vulnerable to a confirmation attack, so NameCoin's protection is technically stronger in terms of client request privacy. The other approaches expose the contents of the queries and replies to the operators; query name minimization (not shown) can be used to limit which servers get to learn the full query. However, clients have no assurance that query name minimization is actually deployed.

DNSSEC intends to protect zone information against zone walks, but all approaches (NSEC1-NSEC4) using offline private key cryptography turned out to be inadequate. The situation is not easily remedied, as NSEC5 (Goldberg et al., 2014) provides an impossibility result showing that online private key operations are necessary to support NXDOMAIN responses, and preventing bulk acquisition of zone data. The proposed scheme

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018

Table 1 – Comparison of the defenses offered by the various designs and their relative deployment complexity.

	Protection against						Ease of migration/ compatibility
	Manipulation by MiTM	Zone walk	Client observation		Traffic amplification	Censorship/legal attacks	
			Network	Operator			
DNS	✘	✓	✘	✘	✘	✘	+++
DNSSEC	✓	✘**	✘	✘	✘	✘	+*
DNSCurve	✓	✓	✓	✘	✘	✘	+*
DNS-over-TLS	✓	n/a	✓	✘	✓	✘	+
Confid. DNS	✘	n/a	✓	✘	✘	✘	++
Namecoin	✓	✘	✓	✓	✓	✓	–
GNS	✓	✓	✓	✓	✓	✓	– –
RAINS	✓	✘	✓	✘	✓	✘	– –

*EDNS0 is not perfectly compatible ** with NSEC5: ✓.
✘: not satisfied, ✓: satisfied, n/a: unchanged from DNS/DNSSEC, "+++/++/+" easy, "– –/–" hard.

for NSEC5 uses two different public keys to separate the offline key used to sign zone data from the online key used to generate NXDOMAIN responses. This way, compromising the online key only enables zone enumeration, but does not impact integrity. In contrast, GNS does not use private keys online or any direct interaction with the zone's authority. GNS can store even confidential data in the name system, effectively protect it from illicit observation by the network or service operators and use offline signing, but cannot support NXDOMAIN. Finally, NameCoin deliberately made the opposite design choice and exposes the full database to all participants.

Using unsolicited DNS replies by open resolvers for traffic amplification is a well-known vector for DDoS attacks. The increased size of DNSSEC responses makes the situation worse, while caching of NSEC replies could also help reduce traffic. Some of the new approaches are not based on UDP, thus making it significantly more difficult to abuse DNS for traffic amplification.

Only the alternative approaches, Namecoin and GNS, are resistant to censorship. Approaches using traditional DNS registrars are inherently vulnerable to legal attacks where influential entities force registrars to block names.

Complete rewrites, especially radical ones like RAINS and GNS, result in cleaner architectures with reduced implementation complexity, but have a significant issue in terms of migration cost compared to more incremental approaches.

11. Conclusion

Technology reflects political philosophies and in turn imposes them upon us. The different name system designs exemplify this situation. The IETF-models around DNS follow a globalist philosophy as they assume that global governance is desirable and possible. SCION follows a more nationalist approach, stressing diversity and isolation. NameCoin follows libertarian ideas of unregulated capitalism, while GNS clearly follows anarchistic ideals.

Pushing for any change in technology to support a particular political philosophy is a difficult task. Modifications to a critical system like DNS, following the general ossification trend of the Internet, are met with inertia. Who will dare to make significant changes which could result in malfunction, impact somebody's business model or a nation state interest?

Acknowledgment

We thank Laura Poitras, Ludovic Courtès, Dan Bernstein, Luca Saiu and Hellekin Wolf for their help and support in preparing this report. We thank William Aiello and the anonymous reviewers for constructive comments.

REFERENCES

A. (NSA). There is more than one way to quantum; 2014. Available from: https://www.documentcloud.org/documents/1076891-there-is-more-than-one-way-to-quantum.html #document/p1. [Accessed 30 October 2017].

Arends R, Austein R, Larson M, Massey D, Rose S. DNS security introduction and requirements, IETF RFC 4033; 2005. Available from: https://tools.ietf.org/html/rfc4033. [Accessed March 2005].

Arnbak A, Goldberg S. Loopholes for circumventing the constitution: unrestrained bulk surveillance on Americans by collecting network traffic abroad. Michigan Telecommun Technol Law Rev 2015;21:317–61.

Bau J, Mitchell J. A security evaluation of DNSSEC with NSEC3. In: Proceedings of the Network and Distributed System Security symposium, NDSS. 2010.

Bernstein DJ. Curve25519: new Diffie-Hellman speed records. In: Public Key Cryptography (PKC). Springer-Verlag LNCS 3958; 2006.

Bernstein DJ. DNSCurve: Usable security for DNS; 2008. Available from: http://dnscurve.org/. [Accessed 30 October 2017].

Bhargavan K, Delignat-Lavaud A, Pironti A, Langley A, Ray M. Transport Layer Security (TLS) session hash and extended master secret extension, RFC 7627 (proposed standard); 2015. Available from: http://www.ietf.org/rfc/rfc7627.txt. [Accessed September 2015].

Bortzmeyer S. Possible solutions to DNS privacy issues; 2013. Available from: http://tools.ietf.org/html/draft-bortzmeyer-dnsop-privacy-sol-00. [Accessed 30 October 2017].

Bortzmeyer S. DNS privacy considerations, RFC 7626 (Informational); 2015. Available from: http://www.ietf.org/rfc/rfc7626.txt. [Accessed August 2015].

Bortzmeyer S. Next step for DPRIVE: resolver-to-auth link; 2016a. https://tools.ietf.org/html/draft-bortzmeyer-dprive-step-2-01. [Accessed 30 October 2017].

Bortzmeyer S. DNS query name minimisation to improve privacy, RFC 7816 (experimental); 2016b. Available from: http://www.ietf.org/rfc/rfc7816.txt. [Accessed March 2016].

Cox R, Muthitacharoen A, Morris R. Serving DNS using a peer-to-peer lookup service. In: Revised papers from the first International workshop on Peer-to-Peer Systems, IPTPS '01. London, UK: Springer-Verlag; 2002. p. 155–65.

Ermert M. DNSSEC-schlüsseltausch 2017 – die vorbereitungen laufen; 2016. Available from: https://www.heise.de/security/meldung/DNSSEC-Schluesseltausch-2017-die-Vorbereitungen-laufen-3273136.html. [Accessed 30 October 2017].

Eudes Y, Grothoff C, Appelbaum J, Ermert M, Poitras L, Wachs M. Morecowbell – nouvelles révélations sur les pratiques de la nsa; 2015. Available from: http://www.lemonde.fr/economie/visuel/2015/01/24/cowbells-nouvelles-revelations-sur-les-pratiques-de-la-nsa_4561547_3234.html. [Accessed 30 October 2017].

Felten EW, Schneider MA. Timing attacks on web privacy. In: Proceedings of the 7th ACM conference on Computer and Communications Security, CCS '00. New York, NY, USA: ACM; 2000. p. 25–32. doi:10.1145/352600.352606.

Goldberg S, Naor M, Papadopoulos D, Reyzin L, Vasant S, Ziv A. Nsec5: provably preventing DNSSEC zone enumeration. IACR Cryptol ePrint Archiv 2014;2014:582.

Herzberg A, Shulman H. Fragmentation considered poisonous: or one-domain-to-rule-them-all.org. In: IEEE conference on Communications and Network Security (CNS). IEEE; 2013. p. 224–32.

Hoffman P, Wijngaards W. Elliptic curve digital signature algorithm (DSA) for DNSSEC, RFC 6605 (proposed standard); 2012. Available from: http://www.ietf.org/rfc/rfc6605.txt. [Accessed April 2012].

Holz R. Empirical analysis of public key infrastructures and investigation of improvements, Ph.D. thesis, TU Munich (submitted December 2013).

Hu Z, Zhu L, Heidemann J, Mankin A, Wessels D, Hoffman P. Specification for DNS over Transport Layer Security (TLS), RFC 7858 (Proposed Standard); 2016. Available from: http://www.ietf.org/rfc/rfc7858.txt. [Accessed May 2016].

I. A. Board. IAB statement on Internet confidentiality; 2014. Available from: https://mailarchive.ietf.org/arch/msg/ietf-announce/ObCNmWcsFPNTIdMX5fmbuJoKFR8. [Accessed 30 October 2017].

Kraft D. Namecoin; 2017. Available from: https://namecoin.org/. [Accessed 30 October 2017].

Krishnan S, Monrose F. DNS prefetching and its privacy implications: when good things go bad. In: Proceedings of the 3rd USENIX conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and more, LEET'10. Berkeley, CA, USA: USENIX Association; 2010.

Mankin A, Wessels D, Heidemann J, Zhu L, Hu Z. t-DNS: DNS over TCP and TLS; 2014. Available from: https://ant.isi.edu/tdns/. [Accessed 30 October 2017].

Möller B, Duong T, Kotowicz K. This POODLE bites: exploiting the SSL 3.0 fallback; 2014. Available from: https://www.openssl.org/~bodo/ssl-poodle.pdf. [Accessed 30 October 2017].

N. T. O. C. (NTOC). Bad guys are everywhere, good guys are somewhere!; 2014. Available from: http://www.spiegel.de/media/media-34757.pdf. [Accessed 30 October 2017].

Nakamoto S. Bitcoin: a peer-to-peer electronic cash system; 2008. Available from: http://bitcoin.org/bitcoin.pdf. [Accessed 30 October 2017].

Paxson V. An analysis of using reflectors for distributed denial-of-service attacks. SIGCOMM Comput Commun Rev 2001;31(3):38–47. doi:10.1145/505659.505664.

Perrig A, Szalachowski P, Reischuk RM, Chuat L. SCION: a secure Internet architecture, ETH Zurich, 2017.

Ramasubramanian V, Sirer EG. The design and implementation of a next generation name service for the internet. SIGCOMM Comput Commun Rev 2004;34(4):331–42.

Redacted (NSA, S32X). QUANTUMTHEORY; 2014. Available from: https://firstlook.org/theintercept/document/2014/03/12/nsa-gchqs-quantumtheory-hacking-tactics/. [Accessed 30 October 2017].

Rossow C. Amplification hell: revisiting network protocols for DDoS abuse. In: In Proceedings of the 2014 Network and Distributed System Security Symposium, NDSS. 2014.

Swartz A. Squaring the triangle: secure, decentralized, human-readable names; 2011. Available from: http://www.aaronsw.com/weblog/squarezooko. [Accessed 30 October 2017].

van Rijswijk-Deij R, Sperotto A, Pras A. DNSSEC and its potential for DDoS attacks: a comprehensive measurement study. In: Proceedings of the 2014 conference on Internet Measurement Conference, IMC '14. New York, NY, USA: ACM; 2014. p. 449–60. doi:10.1145/2663716.2663731.

Wachs M, Schanzenbach M, Grothoff C. A censorship-resistant, privacy-enhancing and fully decentralized name system. In: 13th international conference on Cryptology and Network Security (CANS 2014). 2014. p. 127–42.

Weaver N. A close look at the NSA's most powerful Internet attack tool, Wired.

Wijngaards W. Confidential DNS; 2014. Available from: http://tools.ietf.org/html/draft-wijngaards-dnsop-confidentialdns-02. [Accessed 30 October 2017].

Why top level domains should not use wildcard resource records; 2015. Available from: https://www.icann.org/groups/ssac/documents/sac-015-en. [Accessed 30 October 2017].

Christian Grothoff is a professor.

Matthias Wachs is a research associate.

Monika Ermert is a journalist.

Jacob Appelbaum is a Ph.D. student.

Please cite this article in press as: Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, Toward secure name resolution on the internet, computers & security (2018), doi: 10.1016/j.cose.2018.01.018