The Internet Protocol Journal

March 2021 Volume 24, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.protocoljournal.org

ISSN 1944-1134

From The Editor

We have just completed the annual *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT). The event was to have been held in Manila in the Philippines, but because of the global pandemic it was held as a "virtual" or online event instead. This change of venue is of course not unique to APRICOT. The year 2020 saw many events cancelled, postponed, or converted to online gatherings. In most cases, the Internet remained a reliable and resilient alternative as more and more organizations and individuals took advantage of various networked conferencing systems. Already several studies have documented how the Internet performed through the pandemic. For example, RIPE Labs published "The Lockdown Effect—Implications of the COVID-19 Pandemic on Internet Traffic," which you can find through your favorite search engine.

Many of the core protocols of the Internet have been updated or otherwise enhanced over the years, particularly protocols that originally did not have security as part of their initial design. Development continues within *The Internet Engineering Task Force* (IETF) to improve all aspects of the Internet Protocol Suite, including novel uses of existing technologies. Our first article is a look at current developments in the *Domain Name System* (DNS).

The first IETF meeting was held in January 1986 with a mere 21 attendees. Thirty-five years later the typical IETF meeting attracts about 1,000 attendees from all over the world and lasts a full week, including the pre-IETF *Hackathon* and *Code Sprint* sessions. During the pandemic, IETF meetings too have been confined to online events, this month in "Virtual Prague."

We don't usually publish opinion pieces in this journal, but with 35 years of IETF development and more than 50 years since the origins of the Internet, this seems like a good time to pause and examine where we are with respect to the overall state of our digital economy. Geoff Huston asks, "What have we done?" in his provocative essay that we hope will inspire you to submit your own views in the form of a Letter to the Editor or perhaps an opinion column of your own.

—Ole J. Jacobsen, Editor and Publisher
ole@protocoljournal.org

DNS Trends

by Geoff Huston, APNIC

We used to think of computer networks as being constructed using two fundamental common infrastructure components: *names* and *addresses*. Every connected device had a stable protocol address to allow all other devices to initiate a communication transaction with it by addressing a data packet to this protocol address. And every device was also associated with a name, allowing human users and human-use applications to use a more convenient alias for theses protocol addresses. By mapping names to protocol addresses, the realm of human use could navigate the services of the network by using symbolic names, while at the level of packets the data flow was directed by the network based on topology information of where these device addresses were located.

But that's 1980s thinking and 1980s network architectures.

Communications architectures have evolved, and today's Internet architecture has, very surprisingly, dispensed with that view of the role of addresses. These days, in no small part because of the exhaustion of the IPv4 address pool, but equally because of an architectural evolution that had to cope with the massive explosion of numbers of devices in networks, we've shifted to a client/server network model where clients initiate connections and servers respond. So now clients don't require a permanently assigned network-wide address. Instead, they can use an address only while it is communicating with a server and pass it back to a shared address pool otherwise. Equally, on the server side we've seen the aggregation of uniquely named service points into service delivery platforms, and the multiplexing function that directs clients into the appropriate service rendezvous point is performed at an application level rather than as an infrastructure function. We're now using the address infrastructure in very different ways than the way we had envisaged in the 1980s. Addresses in today's terms look more like ephemeral session tokens on the client side, and coarse rendezvous points on the server side. It is left to the application level to define the specific client-requested service.

But the architecture of the name space and its use has not been static either. The name infrastructure of the Internet is subject to the same evolutionary pressures, and it is these pressures I'd like to look at here. How is the *Domain Name System* (DNS) responding? This survey has three parts: trust, privacy, and all the other stuff.

Trust

Can you believe what the DNS tells you? The answer is that you probably can't!

Many parties have exploited this obvious failure in the trust model in many ways. The DNS is seen as an overt control channel.

For example, you can block access to a named service if that name does not resolve in the DNS. As a consequence, we've seen the rise of deliberate lies in the DNS where content and services that are categorised as harmful are removed from access by withholding the associated name resolution in the DNS. Numerous open resolvers have turned this filtering of the DNS into a positive attribute, and there are many so-called "clean feed" resolvers that do not resolve a collection of service names where the service is deemed to be harmful or criminal in some manner.[1] [2]

This selective filtering of the DNS is a distinguishing feature in the realm of competitive open resolvers. We've also seen numerous national regimes placing the onus on ISPs to block certain services, and given that addresses are no longer uniquely associated with individual services, the implementation of these national regulations is invariably performed through DNS blocking.[3]

We have also seen exercises to attempt to monetise the DNS, where "no such domain" (`NXDOMAIN`) DNS responses are rewritten to send you to a sponsoring search site through response rewriting.

DNS lies have also been used in the IPv6 transition environment where the DNS records—the protocol addresses—are synthesised to allow you to be steered through an IPv4-IPv6 transitional environment.

The motives of all these exercises may vary, but the result is the same, in so far as the DNS answer is a lie.

Then there are the hostile efforts to replace a genuine response with a lie in order to mislead you, in addition to the technique of response guessing to try to insert a fake response before the "normal" response. You can use this technique in DNS over the *User Datagram Protocol* (UDP) transport as the first UDP response whose query section matches the original query the asker used—whether or not it is the "genuine" response. We have also seen manipulated glue records and even attacks on fragmented packets.[4] The insidious nature of these forms of attack is that they rely on the host system to run quite normally. It's the infrastructure of the name system itself that is being perverted here, and the applications are completely unaware of this manipulation.

The response to this need to detect any form of manipulation of the DNS response that has taken place, and, even better, to withhold the lie from the user, is to add a trust mechanism to the DNS. This trust mechanism takes the form of adding digital signatures to DNS responses. The idea is that a digital signature attached to a DNS response can allow the receiver of the response to be assured that the DNS information is current, that it is authentic, that it has not been manipulated or altered, and that it cannot be repudiated. *Domain Name System Security Extensions* (DNSSEC), the framework for adding digital signatures into the DNS, was some 10 years in the making.

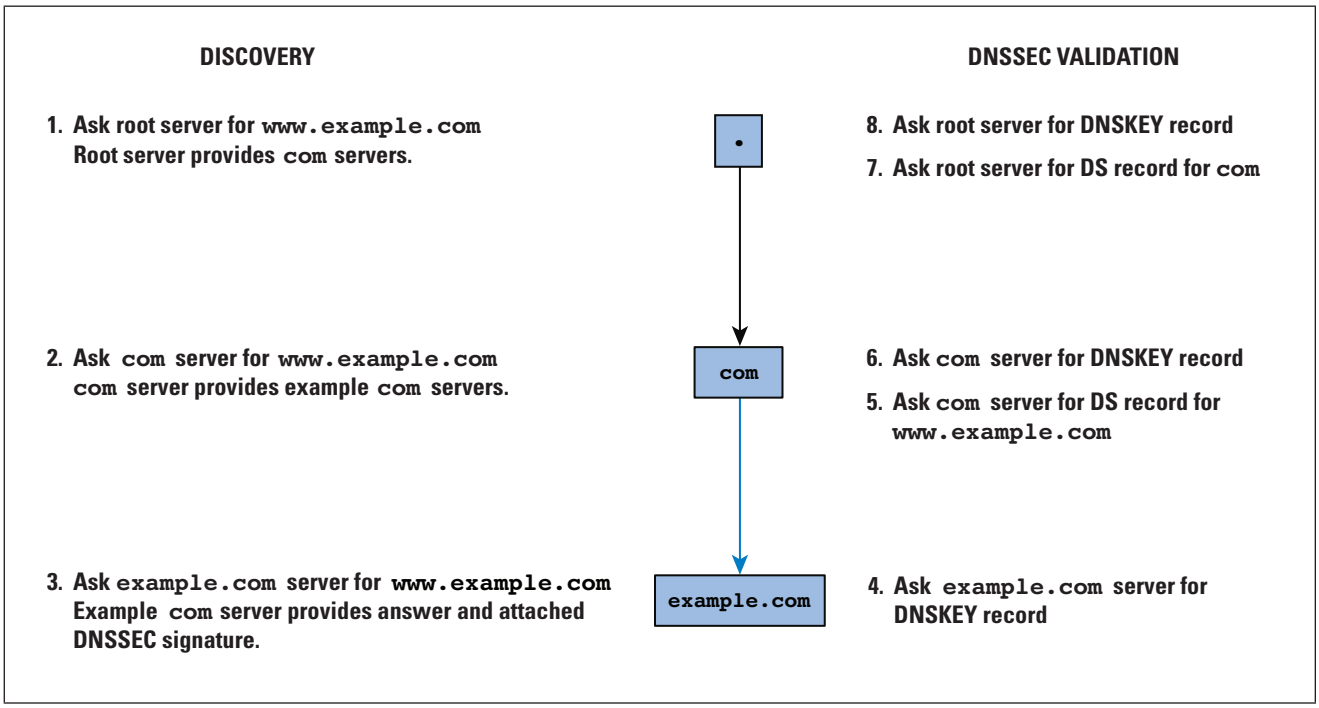
You can associate a key pair (or pairs) with a delegated zone in the DNS, and a set of five further DNS *Resource Records* (RRs) are defined in this framework to sign each entry and to aid in validation of the signature.[5][6][7]

The commentary on DNSSEC deployment varies considerably. Some 25% of the world's users cannot reach a DNS-named service if they cannot validate the DNSSEC signature. That's triple the level from early 2014, so the adoption of validation is certainly gaining some momentum.[8] At the same time, the number of DNSSEC-signed zones appears to be pitifully low. Of the hundreds of millions (perhaps billions these days) of delegated zones in the DNS, we see some 8M signed zones in one such survey.[9] Perhaps a more relevant metric is the ranking of domain names by usage and the ratio of DNSSEC-signed zones in that set. The Alexa top 25 list is a good place to start. None of these is a DNSSEC-signed name.[10] A scan of all `.com`, `.org`, and `.net` second-level domains found that between 0.75 and 1.0% of all domains in these three zones are signed.[11] Zone signing is just not that common in today's DNS.[12] It appears that turning on DNSSEC validation of DNS responses in a recursive resolver has very little downside, given that so few of the popular DNS names appear to be DNSSEC-signed in the first place!

Why is DNSSEC zone signing so uncommon? Are the content and service providers behind these DNS names unconcerned about potential misdirection of users? Of course not! They are extremely concerned because ultimately their potential revenue and reputation are at risk. Are they ignorant about DNSSEC? Again, not at all! Zone signing or not is a choice. These providers want to prevent users from being misdirected, but equally they want to reduce the dependence on intermediaries, and they want your service experience to be as efficient as possible. If DNSSEC was the only choice, then content and service providers would be using it. But it's not the only option. These days service provision uses *Transport Layer Security* (TLS). Almost every service URL out there is an HTTPS URL. Can you be misdirected with TLS? Not normally. Misdirection or deception requires leakage of the service provider's private key, or corruption of the Web *Public Key Infrastructure* (PKI) certificate system. TLS is also fast, because all the credentials needed to validate the certificate are provided in the TLS handshake.

Is DNSSEC validation as fast? Well, no. DNSSEC validation is a serial query sequence all the way back up the name delegation path (Figure 1). Is deployment of DNSSEC zone signing simple? No. There is local key management, the *Zone Signing Key* (ZSK)/*Key Signing Key* (KSK) key split, limited automation, and limited support for high-resilience hosting. And the fundamental criticism is that all this additional effort doesn't stop recursive resolvers from passing back lies in the DNS for DNSSEC-signed zones anyway, because most stub resolvers do not perform DNSSEC validation in any case.

Figure 1: DNSSEC Validation



The entire point is that lies in the DNS were just not possible with DNSSEC. But that assumes that all resolvers perform DNSSEC validation, and that's not the way we've deployed it so far. Many recursive resolvers perform DNSSEC validation. Very few stub resolvers perform DNSSEC validation. This scenario generally works in so far as the recursive resolver withholds the response if the DNSSEC validation fails. But what if the recursive resolver is the one that is telling the lie in the first place? The stub resolver is none the wiser, given that it's not validating, so the lie stands. If the ISP's recursive resolver is blocking some names, performing **NXDOMAIN** substitution, or redirecting actual names, then the stub resolver is just caught in the lie. With all that effort to sign the zone, and all that effort to validate the DNS response, there is absolutely no robust protection against being misdirected. TLS just seems to offer a solution that is faster, simpler, and more robust. No wonder few zone administrators use DNSSEC signing in the DNS service world. It's just a case of more pain, and no real gain.

Is DNSSEC good for anything else? As long as 75% of users sit behind nonvalidating DNS resolver systems—and virtually no users directly validate DNS responses in any case—we cannot place critical information in the DNS securely and expect everyone to be protected by DNSSEC. This means that the incentives for putting critical information into the DNS and protecting it with DNSSEC do not look very convincing. There is simply no natural market-based incentive for deployment of DNSSEC. This conclusion is distressing, because it would certainly be more useful for the network and its captive user population if its name system were trustworthy.

Many have said that the heart of numerous issues with the DNS lies in the choice of a transport protocol for the DNS. The use of UDP as the primary first-choice protocol and the fallback to TCP means that it's challenging to place large quantities of information in DNS answers while still operating within what we've become accustomed to in terms of parameters of speed and robustness.

Validation is a very inefficient process, and the inefficiency is increased by the DNS model where the onus is placed on the client, who is requesting the information, and not the server, who is the source of this information. End clients do not validate because every validation operation would entail further DNS queries in order to construct the validation chain, and the incremental time penalties would be unacceptable in terms of user expectation.

Frustratingly, we know how to make DNSSEC validation faster, and the approach is to pre-provision the validation answers. We can package up all the answers to the DNSSEC validation chain construction queries and include them as additional information to the original signed answer in a single chain extension in the response.[13] However, it's unlikely that this inclusion is viable in a DNS-over-UDP framework. If we want to go down a TLS-like path and package up a validation chain into the DNSSEC-signed response, we will probably have to use DNS over TCP or *DNS over TLS* (DoT).[14] The price of this trust solution is significant, and it creates a higher threshold for the benefits that trusted answers in the DNS can provide. If all this discussion is about protecting users from a Kaminsky-styled attack,[15] then that's just not enough of a case. The benefit needs to be far more than helping justify the considerably higher costs in moving the DNS from UDP to a TCP-styled platform.

Privacy

Everybody looks at the DNS. Everybody. Because the Internet is funded by its users, then what users do on the Internet is of paramount interest to people who sell services to users. Because a lot of crime these days is cybercrime, the criminal and abusive behaviour on the Internet is of fundamental interest to those agencies whose role is to police such behaviours. Because the Internet is now largely about how individuals choose to live their lives and how and why they communicate with others, we've learned that what users do is of paramount interest to government.

How can you find out what users do? Easy. Look at the DNS. Every transaction on the Internet starts with a DNS query, and the DNS exposes every action. But it's worse than that. The DNS is needlessly and senselessly chatty. The DNS overexposes information. These queries and responses are collected, packaged, analysed, profiled, replayed, and traded at all points in the DNS.

How can we make the DNS not the go-to system to expose users and user behaviours to business and government alike? How can we improve its privacy?

There was little in the way of motivation to do anything about this question for years. After all, if the Internet actors are busy constructing a global economy based on surveillance capitalism, why should the parties conducting this surveillance make the task any harder than necessary? The watershed moment that changed the stance for many was the publication of material that Edward Snowden gathered. Government agencies had spent considerable sums in weaponizing the Internet and transforming it into a highly effective surveillance tool that operated at a scale of national populations. Their motivations were not overly concerned about your future purchases, but more about your personal profile. And of all the components of the Internet, the system that laid out all this information in a clear text prepackaged format was the DNS.

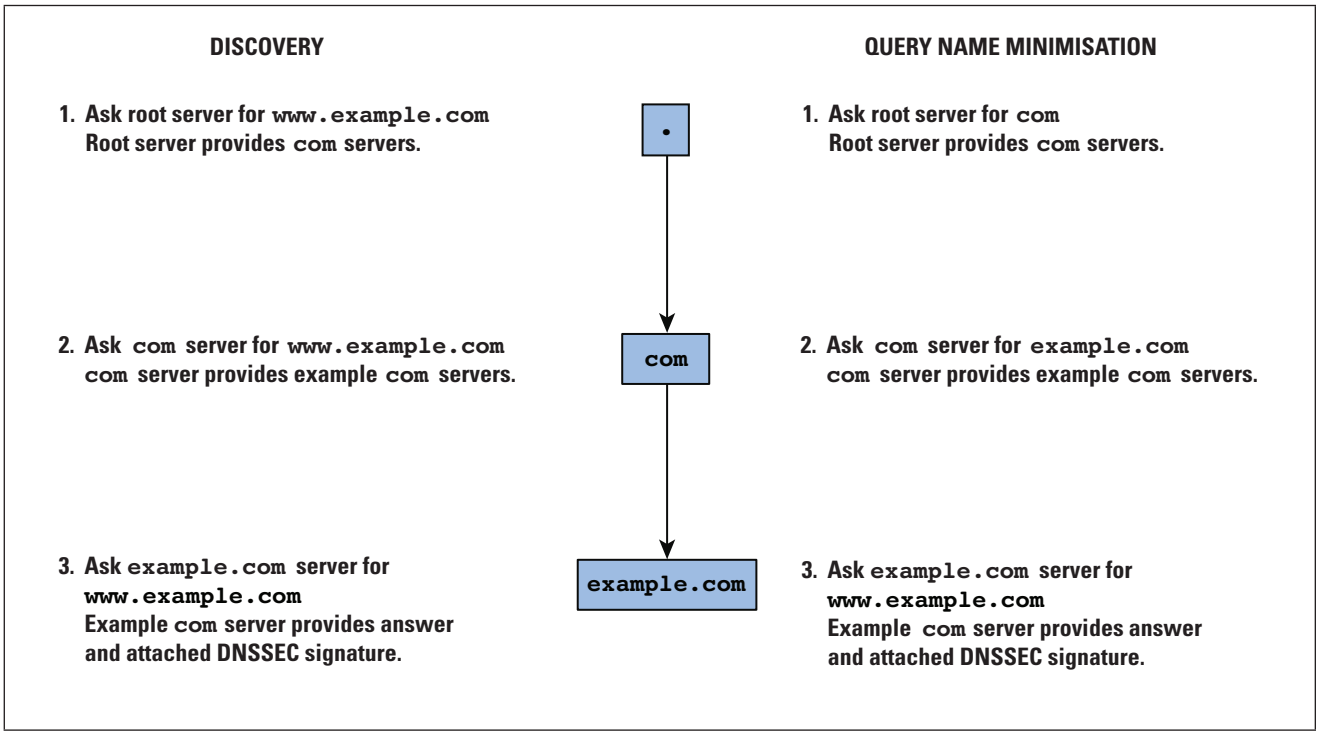
In response, we've been changing aspects of DNS behaviour to try to stop the most blatant forms of information leakage.

The first of this set of privacy-enhancing responses is called *Query Name Minimisation*.[16] The change is to prevent the DNS name-resolution process from being an unconstrained extraneous information leak. This leakage largely relates to the interaction between recursive resolvers and the authoritative name servers. The task of the recursive resolver is to find the right name server to ask, and it starts at the root and asks the query. The response of the root-zone server will direct the queries to the name servers of the relevant delegated top-level domain name, and this process repeats as the resolver traverses down the delegation hierarchy until the resolver has an answer.

But in this process every name server in this sequence, from a root server down, is now aware of the full DNS name that is being resolved. Query Name Minimisation trims the name in these queries so that only the next label is exposed to each name server. Root servers will see only top-level domain name queries, while top-level domain name servers will see only second-level name queries, and so on (Figure 2).

There has been some further work to understand the most robust query type for this discovery process. The initial suggestion of **`NS`** queries has been supplanted by **`A`** queries in the light of experience with this approach. The issue of **`CNAME`** rewriting and the equally vexed question of Empty Non-Terminal domains and the variable behaviour of name servers in such situations have added some complications to this question. This technique of this approach is now widely used, although some implementations have taken some license with the specification and used their own re-interpretation of the technique. Some resolvers, apparently including Google's public resolver service, performs Query Name Minimisation to only the first three levels of the DNS name.

Figure 2: Query Name Minimisation



It appears as if the recursive resolver is deliberately withholding full query name information from the root servers and the top- and second-level domain name services, but is quite willing to disclose the full query name information to servers for zones that are deeper in the name hierarchy. Is this approach motivated by protecting user interests or by an effort to deny information to authoritative servers located at the upper levels of the name hierarchy?

Of course, if we were serious about user privacy, the *Client Subnet* extension would never have been specified.[17] The knowledge of full query names that are emitted by a recursive resolver is to some extent mitigated by the inability to conclusively associate such queries with an end user. But if the query is also loaded with the IP address of the end client, or even the network subnet of the end client, then all pretence of privacy protection has been shredded. While Query Name Minimisation could be seen as a positive step in providing a greater level of concealing extraneous information in the DNS, the use of the Client Subnet value in queries is a gigantic leap backward!

A generic response to privacy considerations on the Internet has been channel encryption. *Telnet* was replaced by *ssh* because of the issues of running sessions over the Internet in the clear. Similarly, **HTTP** has been largely replaced by **HTTPS** for much the same reason. The DNS is increasingly an anachronism in still passing queries and responses in the clear. Not only does it permit eavesdropping, but it also enables efforts to manipulate the responses, all to the detriment of the user.

However, to repeat an earlier observation, the heart of many issues with the DNS lies in its choice of transport protocol. Encryption normally involves many steps, including the presentation and validation of credentials to confirm that clients are talking to the party they intended to talk to, and also to establish a session encryption key to allow encryption of the subsequent data exchange in a manner known to the two parties but unknown to all others. This type of encryption is challenging in UDP. The effort to implement TLS over UDP, namely *Datagram TLS* (DTLS)[18], has the overhead of the exchange of credentials and session cipher establishment, so it's a long step away from a single packet exchange of query and response. DTLS also should avoid IP-level fragmentation, but it cannot avoid large payloads associated with this session establishment process. The result is that fragmentation is pushed up to the application layer and DTLS needs to handle payloads that extend across multiple DTLS datagrams. It appears that the additional overheads of DTLS roughly equate to the overheads of TLS over TCP, but with some added fragility relating to packet fragmentation that is not replicated in TCP. The result of this fragility of DTLS means that when we refer to DNS over TLS, we are in fact referring to *DNS over TLS over TCP* (DoT).[14] It is this TCP-based implementation of TLS that has been implemented and deployed over the path between the stub resolver and the recursive resolver.

DoT adds encryption to the stub-to-resolver path; not only does encryption hide the query and response stream from eavesdroppers, but also DoT prevents alteration or manipulation of the response by third parties. The recursive resolver can still lie about the response, and unless the stub resolver is performing DNSSEC validation (and it's likely not) and the domain name is signed (which it most likely is not), then any DNS lie from the recursive resolver will be unnoticed, whether or not the transport channel from the recursive resolver to the stub resolver uses TLS. A lie is still a lie no matter how secure the packaging used to carry it is. DoT does not eliminate the potential for manipulation of DNS information, but limits the number of entities who are in a position to perform such manipulation and the place and method that the manipulation can be performed. It could be argued that with DoT all you really gain is being better informed as to who is lying to you!

How far should channel cloaking go? Should the identity of the other party be obscured? Should the fact that these transactions are DNS exchanges be obscured? DoT makes no effort to cloak its use. The use of TCP port 853 for DoT is a visible signal that there is an active DoT connection. The use of a novel port number is likely to cause many firewall configurations to trigger their drop filters. The IP address of the remote end is clearly visible, as is the TCP header. The TLS handshake may get around to using *Encrypted Client Hello* (ECH)[19] and encrypt the server name at some point in the future, but in the case of DoT it probably is a minor artefact, given that name-based overloading of service IP addresses is not happening in DoT today and unlikely will in the future.

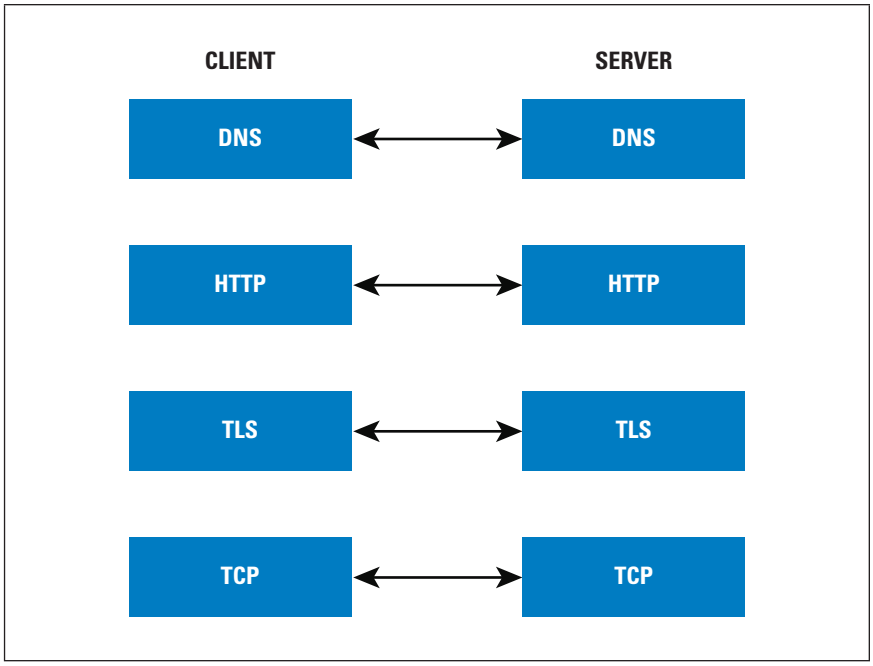
Is DoT going anywhere? It is unlikely in my view. Right now, it requires users to play with their DNS settings, and that is a massive barrier to widespread use. Some users may use it as a means to jump across one set of recursive resolver's DNS filters, such as those provided by their ISP, to hook up with another DNS resolver provider, but with the overt signalling that this is happening, an ISP can readily block this action if it wants to. In theory, the use of TCP permits larger DNS payloads, and we could possibly use *DNS Chaining*[13] to make DNSSEC validation fast and efficient on end systems using DoT. But so many other preconditions, including server provisioning of DNS Chained responses and a reliable way for the DNS to manage large responses, mean that it is still a distant glimmer of a possibility and nothing more.

DoT is seen as a replacement for the existing DNS infrastructure service, where the DNS is a service located on the common platform and applications use the same DNS resolution calls to the platform as they always have. It's a platform approach to securing the DNS. Adoption is probably going to require some form of automated provisioning that typically involves the local access service provider.

Given that the major compromise threat actor here is the same access ISP, and given that the ISP operates the recursive resolver in any case, it's very challenging to understand the incremental benefit of DoT deployment to an ISP. Perhaps it may be that its benefit is as a barrier to other hosts in the local network. Local residential and enterprise environments are cluttered with IP stacks from many providers. A compromised stack is inside the external firewalls and is trusted merely by its physical location. DoT shifts the conversation of a DNS host into a protected channel where the protection is against other hosts on your local network!

DNS over HTTPS (DoH)[20, 27] uses the same TCP and TLS foundations, but adds an HTTP context to the transactions. (Figure 3) A couple of changes here are interesting. The first is the switch to TCP port 443. It looks like any other HTTPS traffic and is not so readily identified in the network as being DNS traffic. Second, the DoH servers do not need to use dedicated IP addresses. Like the web itself, the HTTP protocol allows for named service points. And with TLS 1.3, with ECH you can conceal even the server name in an encrypted envelope. But there is a little more. HTTPS is an application-level protocol, and this approach allows an application to bypass the DNS services the platform provides. Therefore, no ISP-based platform-level configuration is necessarily relevant, and the application can not only conceal its DNS transactions from the local and remote networks, it can also hide these same transactions from the platform and other applications running on the same platform.

Figure 3: DNS over HTTPS (DoH)



If this development heads to DNS over HTTPS/3[21], which uses *Quick UDP Internet Connection* (QUIC)[22, 26], then numerous capabilities are unlocked. Not only is the transport control protocol cloaked behind an encryption envelope in QUIC, but you can make many DNS requests on a single transport channel simultaneously.

How far can we go with this effort to advance a privacy agenda in the DNS? Once we've deployed Query Name Minimisation, discarded Client Subnet, adopted DoH using HTTPS/3 as the application-to-recursive resolver protocol, and pushed DNSSEC validation to the application via attached chained DNSSEC responses, then you have realized much of the achievable trust and privacy agenda. At that point, much of the ability for a third-party onlooker to associate an end-entity identity with a DNS request is severely curtailed, and while a recursive resolver is still privy to these user transactions, the use of DNSSEC all the way to the edge makes response manipulation by any external party, even the recursive resolver itself, particularly challenging when the original DNS data is DNSSEC-signed.

The DNS privacy effort is moving on. The current question is: Should we encrypt the paths between a recursive resolver and authoritative servers? Assuming that Client Subnet has been abolished, there is little that such transactions directly reveal about the identity of the end user, and the larger the pool of clients that a recursive resolver serves, the larger the crowd each individual's queries can hide in. Irrespective of the questions of whether it is feasible (it is) and whether it is scalable (no clear answer, but it looks to have an appreciable incremental cost), the fundamental question of whether channel privacy makes any sense in a privacy context for the individual end user remains.

Other Topics

Other aspects of the technology evolution of the DNS are covered in the following sections.

Internationalized Domain Names (IDNs)

The DNS has traditionally used a 7-bit ASCII code for names. Upper and lower case are equivalent, and in addition to the Latin characters, DNS labels can use hyphens and number characters. In certain circumstances the underscore is also permitted. The expansion of the DNS into a larger character repertoire[23] has not been a stellar success. The design decision was to preserve the capabilities of the DNS system and use encoding to map the larger character set into this restricted alphabet.

The choice of *Unicode*[28] as the underlying character repertoire for this expanded character set was not a very good choice. Unicode involves a contract between an application and a printer. It does not matter that Unicode has multiple ways to print the same glyph on a printer. The printer does not care. But the DNS cares. The DNS has no concept of "what it means," and alternate Unicode strings that are presented in an identical way on a screen actually map to distinct DNS names. So, the effort in the use of Unicode in the DNS has been one of trying to push the Unicode glyph set back into the box and try to specify canonical subsets of Unicode that minimise display similarity. This request is tough, and made even harder by the increasing variance in display glyphs used to display the same Unicode code point. This challenge is most evident in emoji characters.

Why is it a problem? Because the Internet still works on a rather crude model of "what you see is what you get." If alternate ways of coding the same visual outcome are possible, then they are distinct labels in the DNS and can be associated with distinct service points. The possibilities to dupe unsuspecting users is of course a natural and inevitable outcome of this process.

DNS Abuse

These days "DNS Abuse" is a current topic, particularly in the *Internet Corporation for Assigned Names and Numbers* (ICANN) world. The phrase describes an effort to engender a level of self-regulation in the DNS supply industry, where behaviours are governed largely by contractual provisions between the registrant and the registrar, and between the registrar and a common registry. It allows various forms of abusive behaviours, including criminal activities, that use the DNS to be sanctioned by contractual enforcement including takedown of the DNS names. It's a lot like the self-regulatory measures that are common in the finance industry, but without the reporting framework, without any common legal framework for enforcement, and without any penalties for breaches.

My suspicion is that it will turn out to be no more effective than the similar measures to undertake self-regulation in the finance sector, and probably even more ineffectual than the rather unimpressive results that the banking sector has posted. It's unlikely to be successful in reducing the levels of abusive and criminal behaviour that use the DNS and the Internet.

DNS Fragmentation

DNS fragmentation is also a perennial topic in the evolution of the name space. The pressures for a single, consistent name space are embedded in the concept of a single network. Communications systems rely on assumptions of referential integrity, and referential integrity typically implies that the same DNS name refers to the same resource.[24]

We've seen this concept tested many times, from alternate root systems of a couple of decades ago to private name spaces today in the enterprise environment. A good case in point about referential fragmentation is the use of search terms as a replacement for the DNS. The objective of a search engine is to try to customise the responses to best match the known preferences of the querier, and when *I* attempt to pass a pointer to you about a digital resource, the search term that I use that will expose this resource may not be exposed when *you* enter the same search term in your context. This possible difference is not only an attribute of search engines, but a feature.

However, DoH enables other forms of DNS fragmentation. It enables you to lift a name space out of common network infrastructure and place it into the context of an attribute of an application. The application can direct DoH queries to server infrastructures of its own choosing and provide responses that pertain to the application as distinct from a lookup in a common distributed database. The ability in HTTPS to push objects to the application client also allows you to use so-called *Resolverless-DNS*[25], where an application can improve the performance of name resolution functions by performing them in advance of the time they are needed.

Name Flattening

DNS *name flattening* has been a constant pressure in the DNS. Nobody wants to have their critical service names `buried.deep.down.in.the.dns.under.a.bunch.of.other.names`. Not only do such names take longer to resolve, they increase the set of dependencies in the same way because presumably a greater number of service providers all the way down in the name hierarchy exist. DNS users want shorter names. The shorter the better. The result is that the name space is under constant erosive pressure to flatten down. The ultimate place to land is in the top level of the DNS, in the root zone, and as the price premium for top-level domain comes down, the pressure to inflate this zone with significantly larger numbers of entries is an inevitable consequence.

The Future of Names

But perhaps the forces of evolutionary pressure are more fundamental and parallel the evolutionary forces of the Internet itself.

The silicon industry is indeed prodigious, and there are many more processors in this world than people. While we have constructed the DNS name space using an analogue of natural language terms as a means of facilitation of human use, this use pattern is not necessarily the dominant use pattern of the DNS any longer. One view of the DNS today is a universal signalling and tunnelling protocol, and the use of the DNS as a command-and-control channel for malware bot armies testifies to the efficacy of such use of the DNS!

It's likely that as the number of such devices increases, the use of the DNS as an orchestration mechanism increases in importance and the human use of the DNS becomes increasingly marginalised. Human-use DNS may well become an esoteric luxury business. The high-touch activity of DNS name management is unsustainable in a shift from human to largely automated use, and the business models and institutions that populate this space will need to adjust to a names business that provides names not as a branding attribute using natural language tokens, but as an undistinguished commodity activity. In the same way that we have transformed IP addresses from end-point identifiers to ephemeral session tokens, we may well see the DNS as a code base for command and control of highly distributed automated systems, and that is very different from the distributed database lookup that we originally constructed for the human-use model of the DNS.

When we think of a DNS query as a set of instructions to a DNS server, and the DNS server as a distributed processing environment, the DNS changes from a distributed database to a distributed computation and signalling environment. The composition of labels in such a DNS is no longer roughly derived from dictionaries of known words from human languages, but instead is encoded instructions where the labels are in effect a coded program for a name resolver to execute. It is certainly a different future for the DNS as we know it, but its probable commoditisation in the future is in line with the plight of carriage, switching, and content in the Internet!

From this perspective, the evolution of the DNS parallels the larger evolution of the Internet itself, where the infrastructure is not about a human-usable framework any longer, but instead is focussed on providing a highly automated environment where the elements are themselves programs and automata.

That does not mean that the human-use DNS will disappear. But the DNS as we know it today may end up as a small set of high-end luxury boutique activities that make a feature of the luxury of custom procedures to manage persistent names.

In the meantime, the rest of the DNS heads deeper into a commodity utility world of large sets of algorithmically generated transient names that are managed entirely automatically and tailored for one-off use by other processes. It may be that the overwhelming use of tomorrow's DNS has nothing much to do with human names any longer and will be concentrated on serving a largely automated framework that uses the DNS to support a general command-and-control signalling framework. Ephemeral names are as good as, if not better than, persistent names. Registration and attribution processes are largely irrelevant.

The DNS may still be valuable, but individual names will be completely worthless!

References and Further Reading

[1] Quad 9 Open DNS Resolver
https://www.quad9.net/about/

[2] Cloudflare 1.1.1.1 for Families
https://blog.cloudflare.com/introducing-1-1-1-1-for-families/

[3] Young Xu, "Deconstructing the Great Firewall of China," *Thousand Eyes*, March 8, 2016.
https://blog.thousandeyes.com/deconstructing-great-firewall-china/

[4] Kazunori Fujiwara and Paul Vixie, "Fragmentation Avoidance in DNS," November 2020. Internet-Draft, work in progress.
https://tools.ietf.org/html/draft-ietf-dnsop-avoid-fragmentation-03

[5] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends, "DNS Security Introduction and Requirements," RFC 4033, March 2005.

[6] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends, "Resource Records for the DNS Security Extensions," RFC 4034, March 2005.

[7] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends, "Protocol Modifications for the DNS Security Extensions," RFC 4035, March 2005.

[8] APNIC Labs, DNSSEC Validation Report
https://stats.labs.apnic.net/dnssec

[9] DNSSEC Signed Zone Survey
https://www.secspider.net

[10] DNSSEC Name and Shame
https://dnssec-name-and-shame.com

[11] StatDNS website
https://www.statdns.com

[12] Matthäus Wander, "Measurement survey of server-side DNSSEC adoption," 2017 Network Traffic Measurement and Analysis Conference (TMA), Dublin, Ireland, 2017, pp. 1–9, DOI: 10.23919/TMA.2017.8002913.

[13] Paul Wouters, "Chain Query Requests in DNS," RFC 7901, June 2016.

[14] John Heidemann, Duane Wessels, Allison Mankin, Paul Hoffman, and Liang Zhu, "Specification for DNS over Transport Layer Security (TLS)," RFC 7858, May 2016.

[15] Dan Kaminsky, "The Great DNS Vulnerability of 2008," **https://duo.com/blog/the-great-dns-vulnerability-of-2008-by-dan-kaminsky**

[16] Stephane Bortzmeyer, "DNS Query Name Minimisation to Improve Privacy," RFC 7816, March 2016.

[17] Wilmer van der Gaast, Carlo Contavalli, and Warren Kumari, "Client Subnet in DNS Queries," RFC 7871, May 2016.

[18] Eric Rescorla and Nagendra Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347, January 2012.

[19] Christopher Wood, Kazuho Oku, Eric Rescorla, and Nick Sullivan, "TLS Encrypted Client Hello," Internet-Draft, work in progress, March 2021.
https://datatracker.ietf.org/doc/draft-ietf-tlsesni/

[20] Paul Hoffman and Patrick McManus, "DNS Queries over HTTPS (DoH)," RFC 8484, October 2018.

[21] Mike Bishop, Ed., "Hypertext Transport Protocol Version 3 (HTTP/3)," Internet-Draft, work in progress, October 2020.
https://datatracker.ietf.org/doc/draft-ietf-quic-http/

[22] Christian Huitema, Melinda Shore, Allison Mankin, Sara Dickinson, and Jana Iyengar, "Specification of DNS over Dedicated QUIC Connections," Internet-Draft, work in progress, September 2019.
https://tools.ietf.org/html/draft-huitema-quic-dnsoquic-07

[23] John Klensin, "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework," RFC 5890, August 2010.

[24] Internet Architecture Board, "IAB Technical Comment in the Unique DNS Root," RFC 2826, May 2000.

[25] Erik Sy, "Enhanced Performance and Privacy via Resolver-less DNS," August 2019.
https://svs.informatik.uni-hamburg.de/publications/2019/2019-08-13-Sy-preprint-Enhanced_Performance_and_Privacy_via_Resolver-Less_DNS.pdf

[26] Geoff Huston, "A Quick Look at QUIC," *The Internet Protocol Journal*, Volume 22, No. 1, March 2019.

[27] Geoff Huston, "DNS Privacy and the IETF," *The Internet Protocol Journal*, Volume 22, No. 2, July 2019.

[28] Unicode: **https://home.unicode.org/**

[29] George Michaelson, "DoH the right thing," APNIC Blog, February 10, 2021.
https://blog.apnic.net/2021/02/10/doh-the-right-thing/
See also page 24.

[30] Geoff Huston, "DNS Oblivion," APNIC Labs, December 15, 2020.
https://labs.apnic.net/?p=1392

GEOFF HUSTON, B.Sc., M.Sc. A.M., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for building the Internet within the Australian academic and research sector in the early 1990s. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005. He served on the Board of Trustees of the Internet Society from 1992 until 2001. At various times Geoff has worked as an Internet researcher, an ISP systems architect, and a network operator. E-mail: **gih@apnic.net**

Opinion: What Have We Done?

by Geoff Huston

One of the roles of an opinion piece is to challenge your assumptions and present alternative perspectives, and that is certainly what I plan to do here. You may not agree with my views. I'm not even sure that I agree with them all of the time, because some of these opinions are pretty bleak. But if this article provokes you to make your own assessment of the Internet in a broader context of the evolving relationship between society and this technology, then that is perhaps as much as I could ever hope to achieve here. I should also say that I'm writing this opinion piece as an individual and nothing more. I am not pretending to speak on behalf of my employer in any way. These are *my* words and thoughts.

I was asked to speak at an *Internet Governance Forum* during the COVID-cursed year of 2020. I was briefed that "the most useful thing would be to hear your thoughts on what are the big issues at the moment. Where you see things heading. It would mean that you'd be speaking on your areas of interest from your perspective, not necessarily trying to channel some sort of universal Internet zeitgeist."

I have found this brief a challenging one. In some decades of working in this space I've heard many boom-and-bust talks. In addition, I have seen techno-exuberance reach dizzying heights—and then expositions of sobering realities bring it all back to Earth. But behind this phenomenon I have not seen many perspectives that challenge the very fundamentals of the Internet. We appear to assume that the technology is either beneficial, or at worst neutral, and it's the humans in the loop that overreact. Perhaps, even more dangerously, we assume that the technology is competently implemented. This assumption is perhaps the most dangerous one!

My personal view is that we are heading to a Bad Place. A *very* Bad Place.

A Revolution

Compared to our somewhat naive expectations about the role of computers and networking in the 1980's, we have come a long way down a path that now seems to have taken a turn into some dark—and possibly malign—spaces. How else could we have ended up in accusations of rigged elections, "fake" news, and truly bizarre paranoid notions of some form of "deep state" that seems to sit within the collective social psyche these days. But it's not all just a parade of some ridiculous memes that appear to be rooted in human credulity, because we also have to acknowledge the wholesale destruction of livelihoods and the creation of a new technology economy that is based largely on surveillance capitalism. The digital automation of our society has a highly disruptive aspect, and I think we can confidently assert that we are in the middle of a social revolution as fundamental as the industrial revolution. However, in this case we seem to have backed into this one with our eyes closed.

How could we have missed all the signals? Why are we still thinking that the old social contracts are still valid when they are clearly broken? What went wrong? Well, I'm sure exploring that subject would make a great thesis, but we have two problems. Firstly, I have only a few minutes of your time with this article, and secondly, I really don't know why it all went so wrong anyway! So, without truly knowing how it happened, we find ourselves trapped in another massive revolution.

What advice can I offer? Well, if we are talking about social revolutions, then I should say, "Don't trust that Robespierre guy. He's going to kill us all!" Or perhaps, "Napoleon is a genocidal maniac! He hasn't come back from Elbe to make it all better!" But such dire warnings are ineffective because no one listens.

What should I say here?

Perhaps I should simply apologise for my small part in this mess we find ourselves in.

Because it has all turned out so horrendously bad, I think we should have been more aware of the risks, even if at the time they may have sounded totally far-fetched. We said of the Internet: "This is so good everyone should be able to play." And we said: "The Internet is for everyone." But we never really thought about what we really meant when we proclaimed the universality of the Internet. "Everyone should be able to do this?" has turned into "What have we done to ourselves?"

Code

I am probably not a brilliant programmer. In fact, I should admit that I'm a shocking programmer—and I know I'm not the only one. In fact, I'm probably pretty average as a coder. And if that's the average in our profession, then all I can say is that we are all shockingly bad programmers.

We are building these massive edifices of mind-boggling complexity and then replicating all this rather shoddy software in billions of devices. We were told to "move fast and break things," and we did exactly that. We learned to use the end user as the test case. But the consequences are ugly. Your average car has at least 300 processors and huge amounts of code. It mostly works, but just remember that the network that contains the drive control systems also probably contains the entertainment system. *And all this complexity is probably provided by the lowest bidder!* It's cheaper that way. And much riskier. Modern machinery is now at a level of complexity that visibly defies human understanding or control. God alone knows exactly what is in the software-controlled systems on a Boeing 737 Max 8. Boeing apparently does not. Or even the firmware in your fancy digital front doorbell. Bitter experience has taught us that we can turn a few hundred million baby web cams into a massively destructive attack force within seconds.

The *Key Performance Indicators* in our industry are best described by the currently oh-so-fashionable Agile process: "Let's write even crappier code even faster, and let's break more of it!"

Nobody knows how these systems work anymore. Nobody truly understands the dependencies anymore, if they ever did, and the continual stream of software upgrades should give you ample evidence that we are only just bailing out the bilge as fast as we can to stop the entire ship from sinking!

We spend hundreds of millions of dollars on staffing shiny cyber defence bodies to try to show what a great job we are doing to defend ourselves when, in fact, the problem is not the folks who are driving the hostile trucks through the wide-open doors. The real problem is that it's the people just like me who produced the insanely poor code in the first place who left all these gaping holes behind them. Because none of us really is up to the task. And I don't know about everybody else, but I am still on the keyboard. Still writing code. Collectively we have done an amazing job. The Internet is now busted! And it's not clear that we can fix the problem. We can't make it better. Sorry.

Security

It is evident that we have no desire to build truly secure systems. In the rush to digitise our world of services we are taking extraordinary risks. The term "web security" is the punchline to some demented sick joke because the online world is held together by a level of naive trust that makes all other forms of human credulity look restrained and cautious! Even when we thought about what better security might look like, the response was that we have neither the time nor the money to do a better job. We believe that the consumer is so impatient that milliseconds matter far more than security. We continue to cut corners and build fast, faulty code. Maybe we should have said "no" and walked away from the keyboard. But we didn't. Sorry.

We thought we were helping people communicate, because after all, communication is what drives the human experience. We knew that if you change how we communicate you change the nature of human society. We knew that. But we didn't consider that message seriously. None of us envisioned the perversion of that nobly motivated ambition into the incessant deluge of waste products from the social media factory. We only appreciated the role of content mediators when we eliminated them from the planet. This situation is not pretty. We choose to listen only to what we agree with. The Internet has become a vanity-reinforcing gigantic distorted selfie. Sorry.

No Rules

We built this new world so quickly that we outpaced everything else. This new technology has no controls, no regulation, no competition. In the rush to be the first to unleash the ruthless forms of surveillance capitalism on an unsuspecting populace, we have bypassed all the conventional forms of care and restraint. Just seven digital giants dominate our world.

Their unstinting efforts to lobby politicians has turned the political process into a fatally corrupted empty shell. We moved too quickly and no one else kept up.

We wrote our own rules, and Rule Number One was: "Just do it." From Uber to Google the word was "disruption." But the wholesale destruction of the old-world business environment wasn't the worst thing we did. Destruction of retail shopping wasn't the worst thing we did. Far worse is that we privatised the public communications space. We turned our culture and our public discourse into private property. We privatised our intellectual achievements. Who owns antibiotics? Who owns my genetic code? Who owns my personal profile? We turned everything into a transaction. We destroyed our libraries and replaced them with search engines. We replaced journalism with tweets. Our world is no longer a collection of public spaces, but a collection of private enclosures. In some small way, I helped build that reality. Sorry.

No Way Back

Can I provide some helpful suggestions, offer some motivation, or provide some palliative comfort by asserting that our voices matter, and we can change our world for the better? No. I think that we already betrayed you 30 years ago. The glittering prizes that this new technology promised us turned out to be tawdry, corrupted, and debased. We thought technology would be a compelling force for good. We were wrong. I am truly sorry.

The task before us right now is not to make it better. That is way too ambitious. We just can't make it better. There is no way to back out now. Having unleashed these digital monstrosities, we cannot just tie them up again and put everything back into a box. That we cannot do. The best we can do is to somehow accept the terrible situation and the betrayal of trust that got us here and try to deal with it without making it even worse.

Sorry.

—*Geoff Huston,* `gih@apnic.net`

Upcoming Articles in IPJ

"Automatic Disaggregation in the Routing in Fat Trees (RIFT) Protocol," by Bruno Rijsman. RIFT is a new routing protocol being defined in the IETF. This article focuses on one particular feature of RIFT, namely automatic aggregation and disaggregation.

"Network Functions Virtualization (NFV)," by William Stallings. NFV provides a powerful, vendor-independent approach to implementing complex networks with dynamic demands. NFV builds on well-established technologies, including virtual machines, containers, and virtual networks. With the demand from 5G and cloud service providers, as well as enterprises with large internal networks, NFV is becoming an increasingly widespread technology.

Fragments

Postel Service Award Presented to Onno W. Purbo

The Internet Society, a global nonprofit organization that promotes the development and use of an open, globally connected, and secure Internet, recently presented the prestigious *Jonathan B. Postel Service Award* to Onno W. Purbo for his sustained and substantial technical contributions, leadership, and service to the global Internet community.

Named in honor of the technical community legend Jonathan Postel, this award recognizes extraordinary people like Mr. Purbo who have committed themselves to the technological development, growth, and strength of the Internet. Known as "Indonesia's Internet Liberator," Mr. Purbo is a prolific and well-published Internet advocate who has played a key role in democratizing Internet access, making it more affordable especially in Indonesia's rural areas.

"Mr. Purbo's contribution to the digital sector is invaluable and this award marks what he has achieved and inspired others to achieve. His initiative of meaningful Internet access and Community Networks have instilled the growth of not only affordable but accessible Internet in various areas across Indonesia. I am confident this award will embolden others to innovate and follow his steps and overcome the challenges in their communities especially in improving digitalization," said Johnny Plate, Minister of Communication and Information Technology for Indonesia.

Of his many achievements, Mr. Purbo is best known for pioneering the Internet in Indonesia through sophisticated use of wireless and *Voice over Internet Protocol* technologies. He led the first Internet connection at the Institute of Technology in Bandung and used it to build the first Indonesian educational network. He also championed the deregulation of WiFi frequencies and introduced cyber cafes, neighborhood networks, and community cellular networks to Indonesia. Mr. Purbo organized the first community telephony network over Internet and led the re-introduction of ICT into the Indonesian high school curriculum.

Currently, he is involved in the largest Indonesian FREE e-Learning service, which has brought more than 700 courses to nearly 40,000 participants and trained more than 8,000 teachers on e-learning operations.

"It is an honor to receive the highest and priceless acknowledgment given to Indonesia from the Internet communities," said Mr. Purbo.

"With modified simple off-the-shelf gadgets and equipment, one may fulfill the right to access information and knowledge, which is the necessary foundation for any nation to move forward. The Internet Society has acknowledged the approach is one of the right routes towards the Internet for all. The job is indeed not finished. The Postel Service Award sheds light on the way to go for all of us and inspires extraordinary enthusiasm for moving towards a knowledge-based society."

Mr. Purbo was selected by a distinguished international committee comprised of former Postel Award winners which includes Internet visionaries and luminaries. Now in its 21st year, the Postel Award was established in 1999 by the Internet Society to honor individuals and organizations that, through their work, embody the spirit of Jonathan Postel, whose technical influence can be seen at the very heart of many of the protocols which make the Internet work. Andrew Sullivan, President and CEO of the Internet Society, presented the award, which includes a US$20,000 honorarium and a crystal engraved globe, during a virtual ceremony as part of the 109th *Internet Engineering Task Force* (IETF) meeting which took place November 16–20, 2020.

For more information, please visit:
https://www.internetsociety.org.

History of Networking Recordings

Russ White writes: "In 2017, I realized a lot of the people I've worked with over the years were retiring. When these people leave the networking community, they take a wealth of knowledge about the intent, challenges, and inventions of the early Internet. I decided to capture as much of this history in oral format as possible—hence the history of networking recordings were started. I thought, at first, this would be a small, short-lived series, but I have been amazed by the reaction of the community, and the number of technologies and organizations involved in the design and operation of computer networks.

If you know of someone who should be here, please contact me, as I would like to collect as much oral history in this area as I can for this and future generations. These recordings are released under Creative Commons License (CC BY-NC-ND 4.0). This means recordings can be distributed for any noncommercial purposes by anyone, so long as they are released in full (with no modifications)."

The recordings can be found here:
https://rule11.tech/history-of-networking/

NSA Recommends How Enterprises Can Securely Adopt Encrypted DNS

The *National Security Agency* (NSA) recently released a cybersecurity document, "Adopting Encrypted DNS in Enterprise Environments,"[1] explaining the benefits and risks of adopting the encrypted *Domain Name System* (DNS) protocol, *DNS over HTTPs* (DoH), in enterprise environments. The document provides solutions for secure implementation based on enterprise network needs.

DNS translates domain names in URLs into IP addresses, making the Internet easier to navigate. However, it has become a popular attack vector for malicious cyber actors. DNS shares its requests and responses in plaintext, which can be easily viewed by unauthorized third parties. Encrypted DNS is increasingly being used to prevent eavesdropping and manipulation of DNS traffic. As encrypted DNS becomes more popular, enterprise network owners and administrators should fully understand how to properly adopt it on their own systems. Even if not formally adopted by the enterprise, newer browsers and other software may try to use encrypted DNS anyway and bypass the enterprise's traditional DNS-based defenses.

DoH encrypts DNS requests, preventing eavesdropping and manipulation of DNS traffic. While good for ensuring privacy in home networks, DoH can present risks to enterprise networks if it isn't appropriately implemented. The recommendations detailed will assist enterprise network owners and administrators in balancing DNS privacy and governance for their networks. It outlines the importance of configuring enterprise networks appropriately to add benefits to, and not hinder, their DNS security controls. These enterprise DNS controls can prevent numerous threat techniques used by cyber threat actors for initial access, command and control, and exfiltration.

NSA recommends that an enterprise network's DNS traffic, encrypted or not, be sent only to the designated enterprise DNS resolver. This ensures proper use of essential enterprise security controls, facilitates access to local network resources, and protects internal network information. All other DNS resolvers should be disabled and blocked.

NSA seeks to regularly release unique, actionable, and timely cybersecurity guidance to secure the Department of Defense, National Security Systems, and the Defense Industrial Base. For more information or other cybersecurity products, visit:
https://www.NSA.gov/cybersecurity-guidance.

[1] **https://media.defense.gov/2021/Jan/14/2002564889/-1/-1/0/CSI_ADOPTING_ENCRYPTED_DNS_U_OO_102904_21.PDF**

WebRTC Becomes a Standard

The *World Wide Web Consortium* (W3C) and the *Internet Engineering Task Force* (IETF) recently announced that *Web Real-Time Communications* (WebRTC), which powers myriad services, is now an official standard, bringing audio and video communications anywhere on the Web.

WebRTC, comprised of a *JavaScript* API for Web Real-Time Communications and a suite of communications protocols, allows any connected device, on any network, to be a potential communication endpoint, on the Web. WebRTC already serves as a cornerstone of online communication and collaboration services. The WebRTC framework provides the building blocks from which web and app developers can seamlessly add video chat to a range of applications, including tele-education and tele-health, entertainment and gaming, professional and workforce collaboration.

With the foundations standardized and deployed as a royalty-free feature in Web browsers and other devices and platforms, setting up a secure audio-video communication system with WebRTC has become a built-in capability, eliminating the need to install plugins or download separate applications.

WebRTC is massively deployed as a communications platform and powers video conferences and collaboration systems across all major browsers, both on desktop and mobile. Billions of users can interact now that WebRTC makes live video chat easier than ever on the Web. In commercial products and open source projects, WebRTC has vastly expanded the ability to deploy real-time interaction solutions to customers and users.

The year 2020 has shown both how critical WebRTC already is in a world where travel and physical contacts need to be limited, as well as the many improvements that can be brought to the technology to address new usages that have emerged. Organizations are leveraging WebRTC to conduct training, interviews, strategic planning or as a substitute for in-person meetings. Schools and universities have shifted to virtual learning platforms. Families and friends make daily use of products that are built with WebRTC or parts of it.

With the use of WebRTC expanding beyond the initial core design to power video conferences and collaboration systems in web browsers and other ecosystems, more features and more optimizations are now needed. The IETF *WebTransport* work is aiming to build out additional web support for a variety of transport properties. The *WebRTC Ingest Signaling over HTTPS* work is focusing on the development of a protocol to support one-way WebRTC-based audiovisual sessions between broadcasting tools and real-time media broadcast networks. Similar work to expand the use cases of WebRTC is ongoing in the W3C. For more information visit:

https://www.w3.org/TR/webrtc/
https://www.ietf.org/blog/webrtc-milestone/

Nominations Open for Prestigious Internet Hall of Fame

The Internet Society recently announced that nominations are now open for the next *Internet Hall of Fame* class of inductees. The nomination period will close April 23, 2021 and inductees will be announced at an awards ceremony to be held later this year. The Internet Hall of Fame, now in its tenth year, recognizes a select group of visionaries, leaders and luminaries who have made significant contributions to the development and advancement of the open, global Internet.

Through the work of these individuals, including Vint Cerf, Robert Kahn, Leonard Kleinrock, Tim Berners-Lee, and Elizabeth Feinler, among many others, the Internet Hall of Fame reflects the history of the Internet's development and evolution.

"At no point in time has the importance of the Internet and its chief characteristic—to connect—been felt so broadly, and so acutely," said Andrew Sullivan, President and CEO of the Internet Society.

"The critical role the Internet has played throughout the pandemic reinforces now, more than ever, the significance of the people who originally conceived, built, guided and promoted this global network. It is our privilege to highlight their work and contributions."

Individuals worldwide who have played an extraordinary role in the conceptualization, building, and development of the Internet globally will be considered for induction. In addition to those who have been more visible, the Internet Hall of Fame also seeks nominees who have made crucial, behind-the-scenes contributions. Criteria for evaluation include:

> *Impact:* The contribution has made an extraordinary impact on the development or growth of the Internet, and was and may still be directly relevant to the Internet's ongoing advancement and evolution.
>
> *Influence:* The contribution, relative to the Internet, has significantly influenced: 1) the work of others in the field; 2) society at large; or 3) another more defined but critical audience or region.
>
> *Innovation:* The contribution has broken new ground with original thinking/creativity that has established new paradigms, eliminated significant obstacles, or accelerated Internet advancements.
>
> *Reach:* The contribution has significantly impacted the Internet's reach among society at large, within key audiences or specific geographies, with global impact.

Founded in 2012, the Internet Hall of Fame is an ongoing awards program established by the Internet Society to recognize a distinguished and select group of leaders and luminaries who have made significant contributions to the development and advancement of the global open Internet. More information on the program can be found at `http://www.internethalloffame.org/`.

Domain Abuse Activity Reporting

ICANN's *Domain Abuse Activity Reporting* (DAAR) project is a system for studying and reporting on domain name registration and security threat (domain abuse) behavior across *top-level domain* (TLD) registries. The overarching purpose of DAAR is to develop a robust, reliable, reproducible, and replicable methodology for analyzing security threat activity that can then be later used by the ICANN community to facilitate informed policy decisions.

The system collects TLD zone data and complements these data sets with a large set of high-confidence reputation (security threat) data feeds. The aggregated and anonymized data collected by the DAAR system can serve as a platform for studying or reporting daily or historical registration or abuse activity by each registry. The data is currently being pushed to registries using the ICANN *Service Level Agreement Monitoring* (SLAM) system.

The data collected out of the DAAR system is being used to generate the DAAR monthly reports. The reports are point-in-time analysis of all TLDs for which data was available. The report provides aggregated statistics and time-series analysis about security threats of interest to DAAR namely phishing, malware, spam, and botnet command-and-control. For more information visit:
https://www.icann.org/octo-ssr/daar

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. For more information, contact us at **ipj@protocoljournal.org**

Check your Subscription Details!

If you have a print subscription to this journal, you will find an expiration date printed on the back cover. For several years, we have "auto-renewed" your subscription, but now we ask you to log in to our subscription system and perform this simple task yourself. Make sure that *both* your postal and e-mail addresses are up-to-date since these are the only methods by which we can contact you. If you see the words "Invalid E-mail" on your copy this means that we have been unable to contact you through the e-mail address on file. If this is the case, please contact us at **ipj@protocoljournal.org** with your new information. The subscription portal is located here:
https://www.ipjsubscription.org/

Thank You!

Publication of IPJ is made possible by organizations and individuals around the world dedicated to the design, growth, evolution, and operation of the global Internet and private networks built on the Internet Protocol. The following individuals have provided support to IPJ. You can join them by visiting **http://tinyurl.com/IPJ-donate**

Kjetil Aas
Fabrizio Accatino
Michael Achola
Martin Adkins
Melchior Aelmans
Christopher Affleck
Scott Aitken
Jacobus Akkerhuis
Antonio Cuñat Alario
Nicola Altan
Matteo D'Ambrosio
Selva Anandavel
Jens Andersson
Danish Ansari
Finn Arildsen
Tim Armstrong
Richard Artes
Michael Aschwanden
David Atkins
Jac Backus
Jaime Badua
Bent Bagger
Eric Baker
Santosh Balagopalan
Benjamin Barkin-Wilkins
Michael Bazarewsky
David Belson
Hidde Beumer
Pier Paolo Biagi
Tyson Blanchard
John Bigrow
Orvar Ari Bjarnason
Axel Boeger
Keith Bogart
Mirko Bonadei
Roberto Bonalumi
Julie Bottorff Photography
Gerry Boudreaux
L de Braal
Kevin Breit
Thomas Bridge
Ilia Bromberg
Václav Brožík
Christophe Brun
Gareth Bryan
Stefan Buckmann
Caner Budakoglu
Darrell Budic
Scott Burleigh
Chad Burnham
Jon Harald Bøvre
Olivier Cahagne
Antoine Camerlo
Tracy Camp
Ignacio Soto Campos
Fabio Caneparo
Roberto Canonico
David Cardwell
John Cavanaugh
Lj Cemeras
Dave Chapman
Stefanos Charchalakis
Greg Chisholm
David Chosrova
Marcin Cieslak
Lauris Cikovskis
Guido Coenders
Brad Clark
Narelle Clark
Horst Clausen
Joseph Connolly
Steve Corbató
Brian Courtney
Beth and Steve Crocker
Dave Crocker
Kevin Croes
John Curran
André Danthine
Morgan Davis
Jeff Day
Julien Dhallenne
Freek Dijkstra
Geert Van Dijk
David Dillow
Richard Dodsworth
Ernesto Doelling
Michael Dolan
Eugene Doroniuk
Karlheinz Dölger
Joshua Dreier
Lutz Drink
Dmitriy Dudko
Andrew Dul
Joan Marc Riera Duocastella
Pedro Duque
Holger Durer
Mark Eanes
Andrew Edwards
Peter Robert Egli
George Ehlers
Peter Eisses
Torbjörn Eklöv
Y Ertur
ERNW GmbH
ESdatCo
Steve Esquivel
Jay Etchings
Mikhail Evstiounin
Bill Fenner
Paul Ferguson
Ricardo Ferreira
Kent Fichtner
Armin Fisslthaler
Michael Fiumano
The Flirble Organisation
Gary Ford
Jean-Pierre Forcioli
Susan Forney
Christopher Forsyth
Andrew Fox
Craig Fox
Fausto Franceschini
Valerie Fronczak
Tomislav Futivic
Laurence Gagliani
Edward Gallagher
Andrew Gallo
Chris Gamboni
Xosé Bravo Garcia
Osvaldo Gazzaniga
Kevin Gee
Greg Giessow
John Gilbert
Serge Van Ginderachter
Greg Goddard
Tiago Goncalves
Ron Goodheart
Octavio Alfageme Gorostiaga
Barry Greene
Jeffrey Greene
Richard Gregor
Martijn Groenleer
Geert Jan de Groot
Christopher Guemez
Gulf Coast Shots
Sheryll de Guzman
Rex Hale
Jason Hall
James Hamilton
Stephen Hanna
Martin Hannigan
John Hardin
David Harper
Edward Hauser
David Hauweel
Marilyn Hay
Headcrafts SRLS
Hidde van der Heide
Johan Helsingius
Robert Hinden
Asbjørn Højmark
Damien Holloway
Alain Van Hoof
Edward Hotard
Bill Huber
Hagen Hultzsch
Kevin Iddles
Mika Ilvesmaki
Karsten Iwen
David Jaffe
Ashford Jaggernauth
Martijn Jansen
Jozef Janitor
John Jarvis
Dennis Jennings
Edward Jennings
Aart Jochem
Brian Johnson
Curtis Johnson
Richard Johnson
Jim Johnston
Jonatan Jonasson
Daniel Jones
Gary Jones
Jerry Jones
Anders Marius Jørgensen
Amar Joshi
Javier Juan
David Jump
Merike Kaeo
Andrew Kaiser
Christos Karayiannis
David Kekar
Stuart Kendrick
Robert Kent
Jithin Kesavan
Jubal Kessler
Shan Ali Khan
Nabeel Khatri
Dae Young Kim
William W. H. Kimandu
John King
Russell Kirk
Gary Klesk
Anthony Klopp
Henry Kluge
Michael Kluk
Andrew Koch
Ia Kochiashvili
Carsten Koempe
Richard Koene
Alexader Kogan
Antonin Kral
Robert Krejčí
Mathias Körber
John Kristoff
Terje Krogdahl
Bobby Krupczak
Murray Kucherawy
Warren Kumari
George Kuo
Dirk Kurfuerst
Darrell Lack
Andrew Lamb
Richard Lamb
Yan Landriault
Edwin Lang
Sig Lange
Markus Langenmair
Fred Langham
Tracy LaQuey Parker
Rick van Leeuwen
Simon Leinen
Robert Lewis
Christian Liberale
Martin Lillepuu
Roger Lindholm
Link Light Networks
Sergio Loreti

Eric Louie
Adam Loveless
Guillermo a Loyola
Hannes Lubich
Dan Lynch
Sanya Madan
Miroslav Madić
Alexis Madriz
Carl Malamud
Jonathan Maldonado
Michael Malik
Tarmo Mamers
Yogesh Mangar
Bill Manning
Harold March
Vincent Marchand
Gabriel Marroquin
David Martin
Jim Martin
Ruben Tripiana Martin
Timothy Martin
Carles Mateu
Juan Jose Marin Martinez
Ioan Maxim
David Mazel
Miles McCredie
Brian McCullough
Joe McEachern
Alexander McKenzie
Jay McMaster
Mark Mc Nicholas
Carsten Melberg
Kevin Menezes
Bart Jan Menkveld
Sean Mentzer
William Mills
David Millsom
Desiree Miloshevic
Joost van der Minnen
Thomas Mino
Rob Minshall
Wijnand Modderman
Mohammad Moghaddas
Roberto Montoya
Charles Monson
Andrea Montefusco
Fernando Montenegro
Joel Moore
John More
Maurizio Moroni
Brian Mort
Soenke Mumm
Tariq Mustafa
Stuart Nadin
Michel Nakhla
Mazdak Rajabi Nasab
Krishna Natarajan
Naveen Nathan
Darryl Newman
Thomas Nikolajsen
Paul Nikolich
Travis Northrup
Marijana Novakovic
David Oates
Ovidiu Obersterescu
Tim O'Brien
Mike O'Connor
Mike O'Dell
John O'Neill
Jim Oplotnik
Packet Consulting Limited
Carlos Astor Araujo Palmeira
Alexis Panagopoulos
Gaurav Panwar
Manuel Uruena Pascual
Ricardo Patara
Dipesh Patel
Alex Parkinson
Craig Partridge
Dan Paynter
Leif Eric Pedersen
Rui Sao Pedro
Juan Pena
Chris Perkins
Michael Petry
Alexander Peuchert
David Phelan
Derrell Piper
Rob Pirnie
Marc Vives Piza
Jorge Ivan Pincay Ponce
Victoria Poncini
Blahoslav Popela
Eduard Llull Pou
Tim Pozar
David Raistrick
Priyan R Rajeevan
Balaji Rajendran
Paul Rathbone
William Rawlings
Mujtiba Raza Rizvi
Bill Reid
Petr Rejhon
Robert Remenyi
Rodrigo Ribeiro
Glenn Ricart
Justin Richards
Rafael Riera
Mark Risinger
Fernando Robayo
Gregory Robinson
Ron Rockrohr
Carlos Rodrigues
Magnus Romedahl
Lex Van Roon
Alessandra Rosi
David Ross
William Ross
Boudhayan Roychowdhury
Carlos Rubio
Rainer Rudigier
Timo Ruiter
RustedMusic
Babak Saberi
George Sadowsky
Scott Sandefur
Sachin Sapkal
Arturas Satkovskis
PS Saunders
Richard Savoy
John Sayer
Phil Scarr
Gianpaolo Scassellati
Elizabeth Scheid
Jeroen Van Ingen Schenau
Carsten Scherb
Ernest Schirmer
Philip Schneck
Dan Schrenk
Richard Schultz
Timothy Schwab
Roger Schwartz
SeenThere
Scott Seifel
Yury Shefer
Yaron Sheffer
Doron Shikmoni
Tj Shumway
Jeffrey Sicuranza
Thorsten Sideboard
Greipur Sigurdsson
Andrew Simmons
Pradeep Singh
Henry Sinnreich
Geoff Sisson
Helge Skrivervik
Darren Sleeth
Richard Smit
Bob Smith
Courtney Smith
Eric Smith
Mark Smith
Craig Snell
Job Snijders
Ronald Solano
Asit Som
Ignacio Soto Campos
Evandro Sousa
Peter Spekreijse
Thayumanavan Sridhar
Paul Stancik
Ralf Stempfer
Matthew Stenberg
Adrian Stevens
Clinton Stevens
John Streck
Martin Streule
David Strom
Viktor Sudakov
Edward-W. Suor
Vincent Surillo
Terence Charles Sweetser
T2Group
Roman Tarasov
David Theese
Douglas Thompson
Kerry Thompson
Lorin J Thompson
Fabrizio Tivano
Joseph Toste
Rey Tucker
Sandro Tumini
Angelo Turetta
Phil Tweedie
Steve Ulrich
Unitek Engineering AG
John Urbanek
Martin Urwaleck
Betsy Vanderpool
Surendran Vangadasalam
Ramnath Vasudha
Philip Venables
Buddy Venne
Alejandro Vennera
Luca Ventura
Tom Vest
Dario Vitali
Jeffrey Wagner
Don Wahl
Michael L Wahrman
Laurence Walker
Randy Watts
Andrew Webster
Tim Weil
Jd Wegner
Westmoreland Engineering Inc.
Rick Wesson
Peter Whimp
Russ White
Jurrien Wijlhuizen
Derick Winkworth
Pindar Wong
Phillip Yialeloglou
Janko Zavernik
Muhammad Ziad Ziayuddin
Jose Zumalave
Romeo Zwart
Bernd Zeimetz
廖 明沂.

Follow us on Twitter and Facebook `@protocoljournal`

`https://www.facebook.com/newipj`

Call for Papers

The *Internet Protocol Journal* (IPJ) is a quarterly technical publication containing tutorial articles ("What is...?") as well as implementation/operation articles ("How to..."). The journal provides articles about all aspects of Internet technology. IPJ is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. In addition to feature-length articles, IPJ contains technical updates, book reviews, announcements, opinion columns, and letters to the Editor. Topics include but are not limited to:

- Access and infrastructure technologies such as: Wi-Fi, Gigabit Ethernet, SONET, xDSL, cable, fiber optics, satellite, and mobile wireless.
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance.
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping.
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, cloud computing, and quality of service.
- Application and end-user issues such as: E-mail, Web authoring, server technologies and systems, electronic commerce, and application management.
- Legal, policy, regulatory and governance topics such as: copyright, content control, content liability, settlement charges, resource allocation, and trademark disputes in the context of internetworking.

IPJ will pay a stipend of US$1000 for published, feature-length articles. For further information regarding article submissions, please contact Ole J. Jacobsen, Editor and Publisher. Ole can be reached at **ole@protocoljournal.org** or **olejacobsen@me.com**

Supporters and Sponsors

Supporters

Internet Society

cisco

Diamond Sponsors

Google

Ruby Sponsors

ICANN

Sapphire Sponsors

APNIC FOUNDATION

Emerald Sponsors

Afilias

Akamai

APRICOT Asia Pacific Regional Internet Conference on Operational Technologies

COMCAST

DE-CIX

EQUINIX

JPRS JAPAN REGISTRY SERVICES

JUNIPER NETWORKS

lacnic

LinkedIn

linx

netskope

NSRC Network Startup Resource Center

NTT Communications

RIPE NCC RIPE NETWORK COORDINATION CENTRE

TEAM CYMRU

VERISIGN

WIDE PROJECT

Corporate Subscriptions

AFRINIC The Internet Numbers Registry for Africa

amsix amsterdam internet exchange

DNS-OARC

IWL

ISC Internet Systems Consortium

Limelight NETWORKS

qa cafe

For more information about sponsorship, please contact `sponsor@protocoljournal.org`

The Internet Protocol Journal
Link Fulfillment
7650 Marathon Dr., Suite E
Livermore, CA 94550

CHANGE SERVICE REQUESTED

The Internet Protocol Journal is published quarterly and supported by the Internet Society and other organizations and individuals around the world dedicated to the design, growth, evolution, and operation of the global Internet and private networks built on the Internet Protocol.

Email: ipj@protocoljournal.org
Web: www.protocoljournal.org

Printed in the USA on recycled paper.