May 2015
Geoff Huston

Diving into the DNS

If you are at all interested in how the Internet's Domain Name System (DNS) works, then one of the most rewarding meetings that is dedicated to this topic is the DNS OARC workshops. I attended the spring workshop in Amsterdam in early May, and the following are my impressions from the presentations and discussion.

What makes these meetings unique in the context of DNS is the way it combines operations and research, bringing together researchers, builders and maintainers of DNS software systems, and operators of DNS infrastructure services into a single room and a broad and insightful conversation. And its quite a conversation! Perhaps it's the best DNS conversation you could want to have right now.

The first theme of the workshop was the elephant in the Internet, namely the highly potent denial of service attacks that combine queries in the DNS with the large pool of zombie open DNS resolvers to create sustained high volume traffic streams that are used to attack victims. These days the queries are directed against the DNS infrastructure and use random subdomain names in the attack. There is no answer to these queries for random names, and this defeats conventional recursive resolver caching. All these queries are passed to the authoritative name servers, and in sufficient volume to take the servers into a degraded mode of operation. So the measures to try and block this traffic was a major topic of discussion.

Kazunori Fujiwara of JPRS described how cached NSEC records of DNSSEC responses can be used by recursive resolvers to infer the authoritative non-existence of domain names without needing to pass the query to an authoritative name server for the zone. Of course if the zone is not DNSSEC-signed, then this approach is not exactly possible. But the approach has application in the issue of referrals to the root nameservers, as the NSEC records in the root zone can be used to generate an authoritative response of the non-existence of a domain name without actually passing the query to a root name server.

Marek Majkowski of Cloudflare presented on Cloudflare's approach to coping with DNS query floods. An interesting observation in this presentation was that there was little to be achieved in having the DNS server drop a DNS query as most of the server's work was performed in receiving and parsing the incoming packet, and subsequently dropping it was little different from answering it in the larger scheme of things when looking at authoritative server load. One way to address this is to drop faster and drop cheaper, and this presentation described a Cloudflare technique of using the iptables approach and detect the packet drop much earlier in the packet's progress within the server. This iptables approach handles some 1.2M packets per second according to this presentation. They use the BPF module for packet matching against the attack queries and drop matching queries. To achieve even greater throughput for packet scanning they use "floodgate" which offloads this process to an outboard processor in a network interface card which can handle packet inspection and selective drop at a rate of some 6Mpps. Their next step was to automate the process of loading packet filter rules into these filter units using an automated process of detecting anomalous loads according to classification heuristics and triggering a filter rule when this occurs. It is perhaps alarming to see the extent to which folk such as Cloudflare are working to protect their customers against these forms of attack. Game

theory suggests that if you can push costs to the attacker and away from the defender then you have an effective defence strategy. The DNS DDOS attacks appear to leverage the massive dross of cheap and stupid equipment that litters the Internet and turns these units into attackers. Defending against such volumes involves high levels of engineering skill, customized hardware and a certain amount of sheer ingenuity. All this is making defence more expensive. And that's alarming, as this is an escalation thjat places ever greater pressure on the defender. Ultimately in such a situation the defender loses, unless they can shift the incremental burden back to the attacker.

A similar presentation by Matt Weinberg and Piet Barber of Versign on DDOS mitigation strategies noted that the basic defence strategy was to increase the capacity of their network and server system so that they could still respond to genuine queries during an attack. They are also looking at the rate limiting and deep packet inspection approaches, but the basic observation appears to be consistent: defence costs are increasing and the attack costs are not. Sadly, right now it looks like the good guys are not really winning here.

Ralf Weber of Nominum presented on profiles of DDOS data. He had some measurements to suggest that the random subdomain attack intensity was abating in 2015, but the open DNS proxies still continue to be a painful vulnerability in the DNS landscape. Attacks appear to enlist thousand of open resolvers from a larger pool of some 17 million open resolvers. As Ralf points out, outbound rate limiting protections works great for non affected traffic, but it does not protect the attacked domain. For that we need to turn to Ingress list based filtering. A slightly different perspective on DNS data was noted by Bruce Van Nice of Nominum in a day in the life of a resolver, where some 12% of queries appeared to be malicious, while the reamining 88% of queries were "normal" DNS queries. In some ways this is reassuring, in that DNS attacks are not the new "normal" for resolvers. But, unfortunately, its still early days, and the lesson from email, where the amount of spam is now in excess of 98% of all mail, may yet still apply to the DNS.

Florian Maury of ANSSI reported on an query form that sets up a query loop by forcing the resolver to query a malicious authoritative server. Query loops in the DNS are not unknown in the DNS (RFC1034), and the mitigation for this form of attack is for the recursive resolver to limit the number of queries or amount of time it will spend in attempting a resolution.

Cathy Almond of ISC considered an alternate form of rate limiting of recursive queries, looking at threshold SERVFAIL messages being generated by the recursive resolver if the query rate per zone or per server exceeds what is set to be an attack threshold. Its certainly a lighter weight response to random subdomain attacks that automation of specific zone filters in response to each attack, but I can't help but wonder if the SERVFAIL response only encourages the coopted open resolver to repeat the query to other authoritative servers in the zone.

Stephen Lagerholm of Microsoft looked at the way resolvers cache "negative" information. When a name does not exist the authoritative name server will pass back an NXDOMAIN response to indicate that non-existence. It makes sense for recursive resolvers to cache this response, so that repeated queries for the same name can be answered by the recursive resolver without involving the authoritative server. But if this non-existence is the result of a temporary configuration error, or others temporary forms of interruption, then it makes sense to use a shorter cache so that the name can be quickly re-instated when the problem is corrected. For almost one half of the popular domain names they found that the cache time for these negative responses is somewhere between one hour and one day.

A second theme in the workshop was DNSSEC, including investigations into aspects of DNS behaviours as they relate to the proposed roll of the root key of the DNS, as well as reports on how to perform efficient on-the-fly DNSSEC signing systems. I reported to the workshop on the level of support of ECDSA as a signing protocol in DNSSEC. Most of the cryptography used in digital systems uses prime number manipulation, and what protects much of this function is the difficulty of performing prime number factorization. As computers get more powerful the algorithm needs to head

to ever larger numbers and the signatures get bigger. This increases the size of DNS responses when including DNSSEC signatures, and as this happens we run into all kinds of issues with UDP packet fragmentation, TCP fall back and similar. The Elliptical Curve cryptography function uses a different number property associated with the parameters of an elliptical curve. ECDSA can be far smaller than RSA, but is it widely supported? The presentation noted that 1 on 5 users who used RSA-validating resolvers did not validate the response when presented with a zone signed by ECDSA.

Duane Wessels of Verisign presented on traffic effects of changing root zone keys. Five years ago the root zone of the DNS was signed. There are two keys used: a Zone Signing Key, which is rolled every quarter, and a Key Signing Key which has not changed since the original signing back in 2010. Duane describes an exercise in taking a 10 minute query snapshot of queries captured at A-root instances (some 23 M queries) and run these queries through a test rig that can alter the size parameters of the two keys. The potential issues here is that the larger key sizes lead to larger responses in the DNS. This interacts with the profile of EDNS0 UDP buffer size settings, and the larger response sizes would increase the number of TCP sessions seen at the root servers to some extent to what Duane described as a "modest" level. Interestingly, he pointed to some scenarios of key sizes that lead to an increase in the response traffic levels by 35%.

Filippo Valsorda of Cloudflare presented on Cloudflare's impressive work on on-demand DNSSEC-signing of responses. They use dynamically generated zones as they respond to DNS requests with geo-loc content location responses, as distinct from zone-specific data, and want to use NSEC signing in such a manner that prevents zone enumeration. They use a Go implementation of ECDSA that is an astounding 21x faster than the Open SSL C implementation (https://blog.cloudflare.com/go-crypto-bridging-the-performance-gap/). The advantage of ECDSA is that its signatures are small: ECDSA256 is less than a third the size of RSA 1024. They also change the no such name response (NXDOMAIN) which would normally contain 2 NSEC records to a NOERROR response which claims that the requested name exists, but contains just an NSEC record pointing to a synthetic successor record. The result is a 363 octet DNS response which is efficient to compute. They use a novel approach to a missing type response, and use a single ZSK and KSK for all domain response. The result is a highly efficient approach to hosting a large collection of zones at scale.

The workshop also had presentations on tools, techniques and data analysis.

Bert Hubert of PowerDNS gave a very interesting presentation on dnsdist. This tool started life as a simple query distributor that listened on one channel and forwarded queries to one or more resolvers on other channels. What started as simple round robin distribution turned into load balancing with blocking, shunting and shaping policies as well. The observation Bert made was that because recursive resolvers cache their responses a recursive resolver is efficient when its busy, so the load balancer should strive to maintain the activity levels on slave resolvers, keeping as few resolvers as possible as busy as necessary, which is, in effect, a "concentrating loader", as distinct from a generic "load balancer". This is the reverse of most other load balancing functions. The observation is that as they increased the capability of this front end, the lines between this function and a resolver with forwarders started to blur. It's an interesting approach to operating large resolver farms. What fascinates me about PowerDNS is that they are not just another DNS resolver. They appear to be prepared to think carefully about what they are doing and how and come up with some effective and novel solutions that appear to be well tailored to the needs of today's DNS.

Shumon Haque of Verisign looked at the work on qname minimization in the DNS, which attempts to pass only the minimal amount of information to each authoritative name server in the top-down walk of the DNS name hierarchy to resolve a DNS name. Conventional DNS thinking says that if you get a NXDOMAIN at any point in this top down iterative process you then stop. And for the most part this appears to work. Where it fails is when there is a CDN being used that performs DNS mapping via CNAMES. The presentation showed two examples where a fully qualified name query to an intermediate name in a CDN system produces a NS referral to the child Zone, while a query for the exact intermediate name form produces an NXDOMAIN response. This is a worrisome example

where the DNS is starting to behave like the larger Internet: there is DNS "middleware" that performs name transforms on queries that do not work in a consistent manner, and qname minimization exposes such differences. In the case of the DNS the number of actors is still sufficiently small, and the motivation to resolve these behavioural inconsistencies sufficiently high, that the content data networks have treated this as a bug in their code and altered their behaviour to support qname minimization. But will this always be the case?

John Dickinson of Sinodun presented on updates to a DNS monitoring tool, Hedgehog. The task of monitoring the load across a few hundred anycxast instances of a DNS server can be a significant operational challenge, and Hedgehog addresses this is a very straightforward manner. One observation that caught my interest was in a couple of screenshots where he showed a monitor of IPv4 UDP queries vs IPv6 UDP queries. The ratio between the two protocols was 2,000 to 1. It's not clear to me why IPv6 appears to be so little used in the DNS today.

Patrick Wallstrom reported on zonemaster, a zone checking tool that builds upon the old DNScheck and zonecheck tools.

Joao Damas of Bond Internet Systems looked at client to resolver traffic, extracting a profile of what users actually ask of the DNS and the TTLs provided in responses.

Ed Lewis of ICANN talked about the current work on rolling the DNS root Key Signing Key. This is its own unique and difficult problem and I'd like to consider this separately in a later article.

My Impressions

The turning of the DNS from a distributed database query tool into a malicious weapon in the cyber warfare arena has had profound impacts on the thinking about the DNS.

I remember hearing the rallying cry some years back: "Lets all work together to find all these open resolvers and shut them down!" These days I don't hear that any more. It seems that, like SPAM in email, we've quietly given up on eradication, and are now focusing on how to preserve service in a toxic world. I suppose that this is yet another clear case of markets in action – there is no money in eradication, but there is money in meeting a customer's requirement to allow their service to work under any circumstances. We've changed our self-perception from being the public DNS police to private mercenaries who work diligently to protect the interests of our paying customers. We are being paid to care about the victim, not to catch the attacker or even to prevent the attack.

This means that we have changed our focus in the DNS. We are now interested in methods of improving throughput and capacity on certain authoritative name servers to simply absorb attacks. We are looking at UDP processing paths in kernels, ways we can efficiently sign on the fly, and ways we can perform advanced filtering in resolvers to reject attack packets as quickly and efficiently as possible. All this work is not intended to equip authoritative name servers for conventional traffic, but to allow them to continue to serve conventional traffic in the face of these attacks. We are looking at the DNS protocol itself, and think about the differences between "no such domain" and "no such name" responses in order to push attack traffic out of the concentrated middle of the authoritative server back to the edge of the individual recursive resolvers. We are rethinking negative answers in more generic ways with similar intent to deflect traffic away from the authoritative servers.

One line of thought is that all this makes for a more robust DNS that is better for all. And that would be really great if that's what happens.

But I can't help thinking that the attacks have caused a slightly different response, and a worrisome one. Defence is expensive, and really good defence against these forms of attacks is really expensive. Defending your DNS is now a game that you only win if you can afford to win. I worry that by concentrating on the victim rather than the attacker, as we are being compelled to do, these attacks are

creating a two tier DNS system. One for those who can afford to pay for the highly advanced engineering that allows a service to operate in the most trying and difficult of circumstances, and what's left, which is a third rate toxic DNS wasteland that we've simply given up on. The DNS for the rest of us is vanishing in this toxic mire. And it won't correct itself. The attacks are aimed at defended points, so they increase in intensity in line with the increases in defence levels of the highly defended. So everyone else is more and more vulnerable in the face of this increasing malevolence. Is there a way out of this loop of escalating badness? As good as all these attack deflection techniques are, wouldn't it be good if we could just call up the DNS police? Can we shift our collective focus back to the common good, and shift our focus away from selected potential victims who can afford private protection and instead focus on the attacker and the attacks that they carry out?

Personally I think it would be good to see the tables turned and these DNS attackers exposed and prosecuted as the criminal vandals that they undoubtedly are, but I know I'm dreaming at this point. Contemplating such a response raises a massive set of slightly different questions about how to provide security and stability in an Internet not just dominated by competing private sector interests but built almost entirely on these competing private sector interests. We need to think about the functions and capabilities of private sector markets, how to recognize when and why market failures occur, and the role of the national and regional public sector space. I think I've just invoked the magic term "Internet Governance!"

I don't know about you, but at that point my head explodes, and I start to think about how to improve the filtering capability of authoritative name servers and how to signal domain non-existence in more efficient ways. Yes I know that these are no more than stop gap measures, and they are more palliative in nature than curative, but, as they say, its not necessary to out run the lion, its only necessary to run just that little bit faster than the person running alongside you. If that's your aim too, then the DNS OARC workshops have lots of fast running techniques to share!

My thanks to all the workshop presenters for sharing their knowledge and insights, and to DNS OARC for organizing a really fun couple of DNS days.

Author

Geoff Huston B.Sc., M.Sc., 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 1990's. He is author of a number of Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001. He has worked as a an Internet researcher, as a ISP systems architect and a network operator at various times.

www.potaroo.net

Disclaimer