DNSOP Working Group P. Vixie, ISC
INTERNET-DRAFT D. Dagon, GaTech
<draft-vixie-dnsext-dns0x20-00.txt>
Intended Status: Full Standard
Creation Date: March 17, 2008

Use of Bit 0x20 in DNS Labels to Improve Transaction Identity

Abstract

The small (16-bit) size of the DNS transaction ID has made it a frequent target for forgery, with the unhappy result of many cache pollution vulnerabilities demonstrated throughout Internet history. Even with perfectly and unpredictably random transaction ID's, random and birthday attacks are still theoretically feasible. This document describes a method by which an initiator can improve transaction identity using the 0x20 bit in DNS labels.

1 - Introduction and Overview

1.1. This document explains the special relationship between the question section of a DNS request and the question section of the associated DNS response, and shows how this special relationship can be used to convey information in a way that improves transaction identity, making forgeries more expensive.

1.2. It will be argued that this special relationship, while not mentioned in any DNS specification to date, happens to be almost universally true among authority servers now operating on the Internet, and is extremely valuable for its ability to convey information that improves transaction identity, and ought to be made a part of the DNS standard.

1.3. Implementation experience will be presented, showing the first known use of this method, and the measured performance thereof.

1.4. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2 - Question Sections

2.1. In [RFC1035 7.3], the processing of responses is described, including the step of verifying "that the question section corresponds to the information currently desired". This wording seems to leave open several possibilities which are either unexplored or unexplorable, including for example:

> That a response could be useful if its question is of current interest to the requestor even if it does not match any outstanding question; this is not practical since the question section of the response is used, along with the transaction ID, to demultiplex a particular response and match it to some outstanding request.
>
> That a response could "correspond to" a request in some way other than matching it exactly, implying the possibility of wildcarding, or parent domain inclusiveness, or other inexact matches such as case-folded matching as used by responders to look up DNS data in caches or in authority zones.

In practice, all question sections in responses are exact copies of question sections from requests, even if the zone data and answer section owner names differ in their uppercase/lowercase attributes from

the question section. So while it is theoretically possible for a request's question section to contain the name "www.ietf.org" and a response's question section to contain the name "WWW.IETF.ORG", this has

not been observed, and might not even work reliably.

2.2. Because the question section is always, on today's Internet, copied from the request exactly into the response, there is an opportunity to use the 0x20 bit of any ASCII letter (in the ranges 0x41..0x5A and 0x61..0x7A, e.g., A..Z and a..z) in the question name, to convey information from the requestor, to itself, via the responder. For example, the following question names will be treated as equal by a responder, but can be treated as unequal by a requestor:

```
www.ietf.org
WWW.IETF.ORG
WwW.iEtF.oRg
wWw.IeTf.OrG
```

4 - Transaction Identity

4.1. The demultiplexing strategy recommended in [RFC1035 7.3] does not accurately describe current practice. For example, the "name server bug typically encountered in UNIX system" is neither present nor accepted -- responders who answer from a source address other than the destination of the request will not be heard. Therefore the tuple used to match an incoming RCODE=0 or RCODE=3 response to an outstanding OPCODE=0 question is:

```
<ip address, udp port, dns transaction id, qname, qclass, qtype>
```

4.2. Dan Bernstein showed how to randomize the <udp port>, thus increasing transaction identity and therefore also the cost of forging a response in a way that will fool a requestor and perhaps pollute a cache. However, this technique is not universally deployed, and it relies on either a high number of concurrent udp ports, or a high churn rate on udp ports, either of which can be impractical on high volume embedded name servers.

4.3. Much effort has been expended in trying to make the DNS transaction ID more random and less predictable. Ultimately such efforts are insufficient since with only 16 bits to fight over, a determined attacker can use a purely random attack, or even a constant attack, and theoretically, eventually, statistically speaking, break through the requestor's defenses.

5 - Protocol Changes

5.1. By longitudinally encoding one bit of random information per ASCII letter (in the ranges 0x41..0x5A and 0x61..0x7A, e.g., A..Z and a..z) in the question name, the transaction ID can be effectively lengthened

beyond 16 bits. Harkening back to our previous example, here are the 0x20 bits encoded into these question names:

```
   www.ietf.org  111 1111 111
   WWW.IETF.ORG  000 0000 000
   WwW.iEtF.oRg  010 1010 101
   wWw.IeTf.OrG  101 0101 010
```

As explained in Section 3 above, these bits MUST BE ignored by all responders, and "happen to be" copied from the question section of the request into the question section of the response by all known responders, and thus function as a kind of "covert channel" from the requestor, to itself, via the responder.

5.2. It is strongly urged that the DNS specification be amended to require that the question section from the request MUST be copied, exactly, bit for bit, into the question section of the response. The DNS specification is silent on the matter of altering 0x20 bits in the question name when copying it from the request to the response, so, this change is "within the spirit."

A change to the specification is necessary because while such bit for bit copying "happens to be" nearly universal practice today, we must warn all future responder implementors that the 0x20 bits, while not significant for name matching, are now in use as a "covert channel" by the requestor, to itself.

5.3. Requestor implementing this method should ideally signal an error in their operations log when a mismatch in the 0x20 bits occurs, to help measure global cache poisoning attempts, and to diagnose problems which may be due to DNS middleboxes.

5.4. Requestors should take care to remember the original question name, so that following successful verification of the 0x20-randomized question name, the original can be copied into the response message before the other sections are uncompressed. This is because compression pointers in the answer, authority, and additional section often point back to the question section, with the ugly result of copying the 0x20-randomized bits into the cache, and into subsequent responses which include data from the cache.

5.5. In the event of a question name mismatch where the QID, UDP port number, question type, and question class all match, and the question name mismatch is only in the 0x20 bits, then the response should be discarded, and all addresses belonging to this server should be removed from the SLIST (See [RFC1035 7.2]), and the requestor should continue using other available servers (if any). See also sections 6.2 .. 6.4 below. If any of a zone's authoritative name servers can correctly echo

the randomized 0x20 bits, then the transaction should succeed when one of those name servers is eventually reached.

6 - Implementation and Fallback

6.1. Several popular authoritative DNS implemenations including ISC BIND (versions 4, 8, and 9), Nominum ANS, Akamai AKADNS, Neustar UltraDNS, Verisign Atlas, NLNetLabs NSD, PowerDNS, and DJBDNS were tested. All copied the question name exactly, bit for bit, from the request into the response.

6.2. Operational testing has revealed a small set of rare and/or private label authoritative DNS implementations who modify the 0x20 bits in question names while copying the question section from the request to the response. Usually this modification is to set the 0x20 bit, thus converting a domain name to be all-lower-case (0x61..0x7A, e.g., a-z).

6.3. In order to utilize the method described above in section 5.1, before all authoritative DNS servers have implemented the protocol change described above in section 5.2, some kind of fallback strategy must be employed. Such strategy must have a similar security profile to the method described above in section 5.1, or else an attacker could force a victim to discard the benefits of using this method at all.

6.4. One fallback strategy, if all of a zone's authoritative name servers fails to copy the 0x20 bits in the question name from the request to the response, is to repeat the entire sequence, using newly randomized query ID's (and other randomizable query elements, if in use). If possible, the repeated sequence should try the zone's authority servers in a different (random) order each time. If the entire sequence is repeatable several times, where the random QID (and other randomizable query elements, if any) are successfully echoed back each time, then it is reasonable to ignore mismatches in the 0x20 bits.

7 - Security Considerations

7.1. No one knows when the next random number generator weakness will be found, or how long it may take for Secure DNS to be deployed. In fact, no one really knows how many successful transaction ID guessing attacks occur, or how much intentionally polluted cache data exists at any given moment.

7.2. The method described here allows additional transaction identity to

be encoded in a request and verified in a response, thus increasing the cost of DNS cache pollution attacks.

7.3. An unfortunate side effect of this approach is that longer domain names, which contain more 0x20 bits, can encode more transaction identity, and may thus yield better security against forgery and cache pollution. Thus, the domain "www.disney.com" (which has 12 extra bits of transaction identity) is better protected against poisoning attacks than the domain "cia.gov" (which yields only 6 extra bits).

7.4. Authority servers who strip the 0x20 bits from question names when copying from the request to the response will see their query volumes increase by a factor of "several" if the recommendations in section 6.4 above are followed.

7.5. Random number generators will expose more sequential state to outside analysis under this proposal, especially if the recommendations in section 6.4 above are followed. This may give predictive attackers an advantage.

8 - IANA Considerations

There is no work for IANA here.

9 - References

[RFC1035] P. Mockapetris, "Domain Names - Implementation and Specification," RFC 1035, USC/Information Sciences Institute, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP14, March 1997.

---

10 - Authors' Addresses

Paul Vixie (text)

Internet Software Consortium
Redwood City, CA, USA
EMail: vixie@isc.org

David Dagon (idea)

Georgia Institute of Technology
Atlanta, GA, USA
EMail: dagon@cc.gatech.edu

Full Copyright Statement

INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).