Network Working Group
draft-ietf-dnsext-edns0dot5-02.txt

R. Austein
InterNetShare.com, Inc.
H. Alvestrand
Cisco Systems
November 2000

A Proposed Enhancement to the EDNS0 Version Mechanism

Status of this document

Motivation and Scope

EDNS0 [EDNS0] specifies a general framework for extending the packet format used by the Domain Name System protocols. The framework includes a simple version numbering scheme to allow the parties in a DNS protocol exchange to determine which extension features the other party understands. While having the advantage of simplicity, the version numbering scheme as specified has drawbacks:

- It provides no way to deprecate a protocol feature;

- It provides no way to deploy experimental protocol features.

This note proposes to augument the monolithic version numbering mechanism with a mechanism for listing an explicit set of protocol features that a particular implementation supports. We retain version numbering as a way of abbreviating the feature sets that we expect to see in common use.

Model

Our revised extension model for the DNS is designed with three goals in mind:

- We want the protocol to be as simple as possible for the common case of a client or server that implements "mainstream standard DNS";

- We want to provide a safe way to experiment with new protocol features, both inside and outside the deployed DNS;

- We want to provide a safe way to deprecate protocol features.

Our revised extension model has two parts, both of which are carried in the OPT pseudo-RR: the VERSION, which stored in the second octet of the TTL field of the OPT RR, and a variable-length list of FEATURES, stored in the variable part of the OPT RR.

All FEATUREs are extensions of the DNS. We reserve the range of FEATURE numbers from 1 to 100 for describing features of the original RFC 1034/1035 DNS specification that we might eventually choose to deprecate.

Any query/response pair can be described as using a set of DNS FEATUREs. Such features might for instance be:

- Domain binary labels according to [BINARY-LABELS];

- Extended RCODEs (the general principle, not specific values);

- Multi-packet UDP response;

- Increased maximum UDP payload size;

- Character set identification in DNS labels;

- SIG record parsing and checking;

FEATURE numbers are handed out by IANA on a first-come-first-served basis within their appropriate ranges. Any revised specification of a format or function should have its own FEATURE number; in the IETF

process, any significantly changed Internet-Draft should have a new FEATURE number assigned for experimentation.

An assigned VERSION number names a set of FEATUREs. VERSION numbers are assigned by the IETF through a standards action.

Normally, any VERSION number encompasses every FEATURE of all lower VERSION numbers, but the possibility of removing FEATUREs exists for two reasons:

- To remove the need for supporting FEATUREs that turned out to be a Really Bad Idea;

- To allow replacing a badly specified FEATURE with a better specified FEATURE performing the same function that has a new FEATURE number.

Mechanism

We transport explicit feature sets as lists of integers carried in the variable RDATA portion of the EDNS0 OPT pseudo-RR.

The OPTION-CODE for FEATURES is [TBD].

The OPTION-DATA for FEATURES is an ordered list of "feature numbers"; a feature number is represented as a big-endian 16-bit unsigned integer, and the list is sorted into numerically increasing order.

Each feature number names a particular protocol feature that is supported by the implementation that generated this OPT pseudo-RR.

Usage

In most respects, the FEATURE mechanism is used symmetrically by clients and servers; exceptions to this rule are stated after the general explanation.

When composing a DNS message, a client or server includes an OPT record indicating a set of FEATUREs that:

- MUST include all FEATUREs that the client or server believes are relevant to this message;

- MAY include all FEATURES that the client or server is prepared to receive.

This set is expressed as a VERSION and any additional FEATURES required.

In general, we expect that a client or server will include an OPT pseudo-RR that indicates:

- The highest VERSION number that the entity generating the message supports; and

- A small (possibly empty) set of additional FEATUREs not encompassed by the VERSION that the entity deems necessary or desirable.

The above symmetry notwithstanding, we impose one important constraint on the server: while a server is allowed to indicate whatever FEATUREs it believes are relevant or useful, a server MUST NOT make use of any FEATURE in a response that is not within the set of FEATUREs indicated by the client that generated the corresponding request. That is, a response may say "I support FEATURE FOO" regardless of what the client supports, but the rest of the response must not use FEATURE FOO unless the client also supports it.

As a special case, if a client explicitly queries for the OPT RR of the root zone, the server returns an OPT record including all FEATUREs that the server supports. This functionality is provided strictly for diagnostic purposes.

Life Cycle

We expect the life cycle of new features to proceed as follows:

- VERSION X is defined and deployed.

- A new FEATURE is defined and experimentally implemented. All clients and servers taking part in the experiment use FEATURE to indicate support.

- Community consensus is reached that this FEATURE is genuinely useful.

- VERSION X+1 is defined, encompassing all FEATUREs from VERSION X, plus the new FEATURE (and perhaps others).

- The next generation of DNS software supports VERSION X+1, and never use FEATURE.

Risks

While we have tried to provide the ability to deprecate old bad protocol features, such an ability should be used only rarely, if at all, since by any realistic estimate it takes years (decades?) to upgrade all the DNS implementations already in the field.

A flexible extension mechanism of this type increases the risk that some implementors might chose to deploy features designed to hinder interoperability (so-called "labeled noninteroperability").

Security Considerations

We do not believe that this protocol enhancement adds any major new security risks, but we do believe that it would be helpful in getting complicated DNS extensions such as [DNSSEC] deployed more quickly.

As with any enhancement to or complication of the DNS protocol, this enhancement offers attackers yet another way to increase the load on a name server. Root, TLD and other "major" name servers should view excessively complicated FEATURE sets with suspicion, and should not allow themselves to be tricked into performing more work than is really necessary.

IANA Considerations

IANA will need to allocate an EDNS0 option code for FEATURES.

IANA will need to create a new registry of feature numbers.

Acknowledgments

The authors would like to thank the following people for their help in improving this document: Randy Bush, Patrik Faltstrom, Olafur Gudmundsson, Bob Halley, and XXX.

References

[DNSSEC] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

[DNS-CONCEPTS] Mockapetris, P., "Domain names - concepts and facilities", RFC 1034, November 1987.

[DNS-IMPLEMENTATION] Mockapetris, P., "Domain names - implementation and specification", RFC 1035, November 1987.

[EDNS0] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

[BINARY-LABELS] Crawford, M., "Binary Labels in the Domain Name System", RFC 2673 August 1999.

Author's addresses:

Rob Austein
InterNetShare.com, Inc.
505 West Olive Ave., Suite 321
Sunnyvale, CA 94086
USA

sra@hactrn.net

Harald Tveit Alvestrand
Cisco Systems
Weidemanns vei 27
N-7043 Trondheim
NORWAY

+47 73 50 33 52
Harald@Alvestrand.no