Domain Name System Operations
Internet-Draft
Intended status: Informational
Expires: October 10, 2019

A. Dulaunoy
CIRCL
A. Kaplan
CERT.at
P. Vixie
H. Stern
Farsight Security, Inc.

April 8, 2019

Passive DNS - Common Output Format
draft-dulaunoy-dnsop-passive-dns-cof-06

Abstract

This document describes a common output format of Passive DNS Servers which clients can query. The output format description includes also in addition a common semantic for each Passive DNS system. By having multiple Passive DNS Systems adhere to the same output format for queries, users of multiple Passive DNS servers will be able to combine result sets easily.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 10, 2019.

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
2. Limitation 3
3. Common Output Format 3
3.1. Overview 4
3.2. ABNF grammar 4
3.3. Mandatory Fields 4
3.3.1. rrname 5
3.3.2. rrtype 5
3.3.3. rdata 5
3.3.4. time_first 6
3.3.5. time_last 6
3.4. Optional Fields 6
3.4.1. count 6
3.4.2. bailiwick 6
3.5. Additional Fields 6
3.5.1. sensor_id 6
3.5.2. zone_time_first 7
3.5.3. zone_time_last 7
3.5.4. origin 7
3.6. Additional Fields Registry 7
4. Acknowledgements 7
5. IANA Considerations 7
6. Privacy Considerations 7
7. Security Considerations 8
8. References 8
8.1. Normative References 8
8.2. References 9
8.3. Informative References 10
Appendix A. Examples 10
Authors' Addresses 11

1. Introduction

Passive DNS is a technique described by Florian Weimer in 2005 in Passive DNS replication, F Weimer - 17th Annual FIRST Conference on Computer Security [WEIMERPDNS]. Since then multiple Passive DNS implementations were created and evolved over time. Users of these Passive DNS servers may query a server (often via WHOIS [RFC3912] or

HTTP REST [REST]), parse the results and process them in other applications.

There are multiple implementations of Passive DNS software. Users of passive DNS query each implementation and aggregate the results for their search. This document describes the output format of four Passive DNS Systems ([DNSDB], [DNSDBQ], [PDNSCERTAT], [PDNSCIRCL] and [PDNSCOF]) which are in use today and which already share a nearly identical output format. As the format and the meaning of output fields from each Passive DNS need to be consistent, we propose in this document a solution to commonly name each field along with their corresponding interpretation. The format follows a simple key-value structure in JSON [RFC4627] format. The benefit of having a consistent Passive DNS output format is that multiple client implementations can query different servers without having to have a separate parser for each individual server. passivedns-client [PDNSCLIENT] currently implements multiple parsers due to a lack of standardization. The document does not describe the protocol (e.g. WHOIS [RFC3912], HTTP REST [REST]) nor the query format used to query the Passive DNS. Neither does this document describe "pre-recursor" Passive DNS Systems. Both of these are separate topics and deserve their own RFC document. The document describes the current best practices implemented in various Passive DNS server implementations.

1.1. Requirements Language

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 RFC 2119 [RFC2119].

2. Limitation

As a Passive DNS servers can include protection mechanisms for their operation, results might be different due to those protection measures. These mechanisms filter out DNS answers if they fail some criteria. The bailiwick algorithm [BAILIWICK] protects the Passive DNS Database from cache poisoning attacks [CACHEPOISONING]. Another limitation that clients querying the database need to be aware of is that each query simply gets a snapshot-answer of the time of querying. Clients MUST NOT rely on consistent answers. Nor must they assume that answers must be identical across multiple Passive DNS Servers.

3. Common Output Format

3.1. Overview

The formatting of the answer follows the JSON [RFC4627] format. In fact, it is a subset of the full JSON language. Notable differences are the modified definition of whitespace ("ws"). The order of the fields is not significant for the same resource type.

The intent of this output format is to be easily parsable by scripts. Each JSON object is expressed on a single line to be processed by the client line-by-line. Every implementation MUST support the JSON output format.

Examples of JSON (Appendix A) output are in the appendix.

3.2. ABNF grammar

Formal grammar as defined in ABNF [RFC2234]

```
answer          = entries
entries         = * ( entry CR)
entry           = "{" keyvallist "}"
keyvallist      = [ member *( value-separator member ) ]
member          = qm field qm name-separator value
name-separator  = ws %x3A ws            ; a ":" colon
value           = value                 ; as defined in the JSON RFC
value-separator = ws %x2C ws            ; , comma. As defined in JSON
field           = "rrname" | "rrtype" | "rdata" | "time_first" |
                  "time_last" | "count" | "bailiwick" | "sensor_id" |
                  "zone_time_first" | "zone_time_last" | "origin" |
                  futureField
futureField     = string
CR              = %x0D
qm              = %x22                  ; " a quotation mark
ws              = *(
                    %x20 |              ; Space
                    %x09                ; Horizontal tab
                   )
```

Note that value is defined in JSON [RFC4627] and has the exact same specification as there. The same goes for the definition of string.

3.3. Mandatory Fields

Implementation MUST support all the mandatory fields.

Uniqueness property: the tuple (rrname,rrtype,rdata) will always be unique within one answer per server. While rrname and rrtype are

always individual JSON primitive types (strings, numbers, booleans or null), rdata MAY return multiple resource records or a single record. When multiple resource records are returned, rdata MUST be a JSON array. In the case of a single resource record is returned, rdata MUST be a JSON string or a JSON array containing one JSON string. Senders SHOULD send an array for rdata, but receivers MUST be able to accept a single-string result for rdata.

3.3.1. rrname

This field returns the name of the queried resource.

3.3.2. rrtype

This field returns the resource record type as seen by the passive DNS. The key is rrtype and the value is in the interpreted record type represented as a JSON [RFC4627] string. If the value cannot be interpreted, the decimal value is returned following the principle of transparency as described in RFC 3597 [RFC3597]. Then the decimal value is represented as a JSON [RFC4627] number. The resource record type can be any values as described by IANA in the DNS parameters document in the section 'Resource Record (RR) TYPEs' (http://www.iana.org/assignments/dns-parameters). Supported textual descriptions of rrtypes include: A, AAAA, CNAME, etc. A client MUST be able to understand these textual rrtype values represented as a JSON [RFC4627] string. In addition, a client MUST be able to handle a decimal value (as mentioned above) answer represented as a JSON [RFC4627] number.

3.3.3. rdata

This field returns the resource records of the queried resource. When multiple resource records are returned, rdata MUST be a JSON array containing JSON strings. In the case of a single resource record is returned, rdata MUST be a JSON string or a JSON array containing one JSON string. Each resource record is represented as a JSON [RFC4627] string. Each resource record MUST be escaped as defined in section 2.6 of RFC4627 [RFC4627]. Depending on the rrtype, this can be an IPv4 or IPv6 address, a domain name (as in the case of CNAMEs), an SPF record, etc. A client MUST be able to interpret any value which is legal as the right hand side in a DNS master file RFC 1035 [RFC1035] and RFC 1034 [RFC1034]. If the rdata came from an unknown DNS resource records, the server must follow the transparency principle as described in RFC 3597 [RFC3597].

3.3.4. time_first

This field returns the first time that the record / unique tuple (rrname, rrtype, rdata) has been seen by the passive DNS. The date is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). The time zone MUST be UTC. This field is represented as a JSON [RFC4627] number.

3.3.5. time_last

This field returns the last time that the unique tuple (rrname, rrtype, rdata) record has been seen by the passive DNS. The date is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). The time zone MUST be UTC. This field is represented as a JSON [RFC4627] number.

3.4. Optional Fields

Implementations SHOULD support one or more fields.

3.4.1. count

Specifies how many authoritative DNS answers were received at the Passive DNS Server's collectors with exactly the given set of values as answers (i.e. same data in the answer set - compare with the uniqueness property in "Mandatory Fields"). The number of requests is expressed as a decimal value. This field is represented as a JSON [RFC4627] number.

3.4.2. bailiwick

The bailiwick is the best estimate of the apex of the zone where this data is authoritative.

3.5. Additional Fields

Implementations MAY support the following fields:

3.5.1. sensor_id

This field returns the sensor information where the record was seen. It is represented as a JSON [RFC4627] string.

If the data originate from sensors or probes which are part of a publicly-known gathering or measurement system (e.g. RIPE Atlas), a JSON [RFC4627] string SHOULD be prefixed.

3.5.2. zone_time_first

This field returns the first time that the unique tuple (rrname, rrtype, rdata) record has been seen via master file import. The date is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). The time zone MUST be UTC. This field is represented as a JSON [RFC4627] number.

3.5.3. zone_time_last

This field returns the last time that the unique tuple (rrname, rrtype, rdata) record has been seen via master file import. The date is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). The time zone MUST be UTC. This field is represented as a JSON [RFC4627] number.

3.5.4. origin

Specifies the resource origin of the Passive DNS response. This field is represented as a Uniform Resource Identifier [RFC3986] (URI).

3.6. Additional Fields Registry

In accordance with [RFC6648], designers of new passive DNS applications that would need additional fields can request and register new field name at https://github.com/adulau/pdns-qof/wiki/Additional-Fields.

4. Acknowledgements

Thanks to the Passive DNS developers who contributed to the document.

5. IANA Considerations

This memo includes no request to IANA.

6. Privacy Considerations

Passive DNS Servers capture DNS answers from multiple collecting points ("sensors") which are located on the Internet-facing side of DNS recursors ("post-recursor passive DNS"). In this process, they intentionally omit the source IP, source port, destination IP and destination port from the captured packets. Since the data is captured "post-recursor", the timing information (who queries what) is lost, since the recursor will cache the results. Furthermore, since multiple sensors feed into a passive DNS server, the resulting data gets mixed together, reducing the likelihood that Passive DNS

Servers are able to find out much about the actual person querying the DNS records nor who actually sent the query. In this sense, passive DNS Servers are similar to keeping an archive of all previous phone books - if public DNS records can be compared to phone numbers - as they often are. Nevertheless, the authors strongly encourage Passive DNS implementors to take special care of privacy issues. bortzmeyer-dnsop-dns-privacy is an excellent starting point for this. Finally, the overall recommendations in RFC6973 [RFC6973] should be taken into consideration when designing any application which uses Passive DNS data.

In the scope of the General Data Protection Regulation (GDPR - Directive 95/46/EC), operators of Passive DNS Server needs to ensure the legal ground and lawfulness of its operation.

7. Security Considerations

In some cases, Passive DNS output might contain confidential information and its access might be restricted. When a user is querying multiple Passive DNS and aggregating the data, the sensitivity of the data must be considered.

8. References

8.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, DOI 10.17487/RFC2234, November 1997, <https://www.rfc-editor.org/info/rfc2234>.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September 2003, <https://www.rfc-editor.org/info/rfc3597>.

[RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, DOI 10.17487/RFC3912, September 2004, <https://www.rfc-editor.org/info/rfc3912>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <https://www.rfc-editor.org/info/rfc4627>.

[RFC5001] Austein, R., "DNS Name Server Identifier (NSID) Option", RFC 5001, DOI 10.17487/RFC5001, August 2007, <https://www.rfc-editor.org/info/rfc5001>.

[RFC6648] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating the "X-" Prefix and Similar Constructs in Application Protocols", BCP 178, RFC 6648, DOI 10.17487/RFC6648, June 2012, <https://www.rfc-editor.org/info/rfc6648>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

8.2. References

[BAILIWICK] Edmonds, R., "Passive DNS Hardening", 2010, <https://archive.farsightsecurity.com/Passive_DNS/passive_dns_hardening_handout.pdf>.

[CACHEPOISONING] Kaminsky, D., "Black ops 2008: It's the end of the cache as we know it.", 2008, <http://kurser.lobner.dk/dDist/DMK_BO2K8.pdf>.

[DNSDB] Security, F., "DNSDB API", 2013, <https://api.dnsdb.info/>.

[DNSDBQ] Vixie, P., "DNSDB API Client, C Version", 2018, <https://github.com/dnsdb/dnsdbq>.

[PDNSCERTAT]
CERT.at, "pDNS presentation at 4th Centr R&D workshop Frankfurt Jun 5th 2012", 2012, <http://www.centr.org/system/files/agenda/attachment/rd4-papst-passive_dns.pdf>.

[PDNSCIRCL]
Luxembourg, C. -. I. R. C., "CIRCL Passive DNS", 2012, <https://www.circl.lu/services/passive-dns/>.

[PDNSCLIENT]
Lee, C., "Queries 5 major Passive DNS databases: BFK, CERTEE, DNSParse, ISC, and VirusTotal.", 2013, <https://github.com/chrislee35/passivedns-client>.

[PDNSCOF] Dulaunoy, D. P. A., "Passive DNS server interface using the common output format", 2013, <https://github.com/D4-project/analyzer-d4-passivedns/>.

[REST] Fielding, R. T., "Representational State Transfer (REST)", 2000, <http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>.

[WEIMERPDNS]
Weimer, F., "Passive DNS Replication", 2005, <http://www.enyo.de/fw/software/dnslogger/first2005-paper.pdf>.

8.3. Informative References

[I-D.narten-iana-considerations-rfc2434bis]
Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", draft-narten-iana-considerations-rfc2434bis-09 (work in progress), March 2008.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

Appendix A. Examples

The JSON output are represented on multiple lines for readability but each JSON object should be on a single line.

If you query a passive DNS for the rrname www.ietf.org, the passive dns common output format can be:

{"count": 102, "time_first": 1298412391, "rrtype": "AAAA",
"rrname": "www.ietf.org", "rdata": "2001:1890:1112:1::20",
"time_last": 1302506851}
{"count": 59, "time_first": 1384865833, "rrtype": "A",
"rrname": "www.ietf.org", "rdata": "4.31.198.44",
"time_last": 1389022219}

If you query a passive DNS for the rrname ietf.org, the passive dns common output format can be:

{"count": 109877, "time_first": 1298398002, "rrtype": "NS",
"rrname": "ietf.org", "rdata": "ns1.yyz1.afilias-nst.info",
"time_last": 1389095375}
{"count": 4, "time_first": 1298495035, "rrtype": "A",
"rrname": "ietf.org", "rdata": "64.170.98.32",
"time_last": 1298495035}
{"count": 9, "time_first": 1317037550, "rrtype": "AAAA",
"rrname": "ietf.org", "rdata": "2001:1890:123a::1:1e",
"time_last": 1330209752}

Please note that the examples imply that a single query returns a single set of JSON objects. For example, two queries were made; one query returned a set of two JSON objects and the other query returned a set of three JSON objects. This specification requires each JSON object individually MUST conform to the common output format, but this specification does not require that a query will return a set of JSON objects.

Please note that in the examples above, any backslashes "\" can be ignored and are an artifact of the tools which produced this document.

Authors' Addresses

Alexandre Dulaunoy
CIRCL
16, bd d'Avranches
Luxembourg L-1160
Luxembourg

Phone: (+352) 247 88444
Email: alexandre.dulaunoy@circl.lu
URI: http://www.circl.lu/

L. Aaron Kaplan
CERT.at
Karlsplatz 1/2/9
Vienna A-1010
Austria

Phone: +43 1 5056416 78
Email: kaplan@cert.at
URI: http://www.cert.at/

Paul Vixie
Farsight Security, Inc.
11400 La Honda Road
Woodside, California 94062
U.S.A.

Email: paul@redbarn.org
URI: https://www.farsightsecurity.com/

Henry Stern
Farsight Security, Inc.
11400 La Honda Road
Woodside, California 94062
U.S.A.

Phone: +1 650 542-7836
Email: henry@stern.ca
URI: https://www.farsightsecurity.com/