Internet Engineering Task Force (IETF)
Internet-Draft
Intended Status: Standards Track
Expires: May 11, 2015

Phillip Hallam-Baker
Comodo Group Inc.

November 7, 2014

Private-DNS
draft-hallambaker-privatedns-01

Abstract

This document describes Private DNS, a transport security mechanism for the DNS protocol. The mechanism may be employed to secure communication between a client and its resolver or between a resolver and an authoritative server.

Service binding including key exchange is effected using the JSON Service Connect (JCX) Protocol. DNS protocol messages are wrapped in a new framing protocol.

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 http://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."

Copyright Notice

Table of Contents

1. Introduction.

Recent events have required urgent consideration of privacy concerns in Internet protocols. In particular the lack of confidentiality controls in the DNS [RFC1035] protocol is of considerable concern.

This document describes Private-DNS, a security enhancement for the DNS protocol that meets the principal use cases and requirements set out in [I-D.hallambaker-dnse]. This enhancement provides for encryption and authentication of the DNS protocol messages.

Private-DNS makes use of the JSON Service Connect (JCX) Protocol [I-D.hallambaker-wsconnect] and the UYFM framing protocol described in that specification.

1.1. Related Work

The proposal approach compliments the integrity controls provided by DNSSEC [RFC4033]. While both provide integrity controls, the controls provided by DNSSEC are based on digital signatures while this proposal provides controls based on a Message Authentica Code technique.

Like the Omnibroker protocol [I-D.hallambaker-omnibroker], this proposal is built on JCX [I-D.hallambaker-wsconnect] but offers a low level interface to the DNS protocol alone as opposed to a high level interface to generalized discovery services. A client would use the DNSE-JX interface in cases where retrieval of specific DNS resource records is required. The OmniBroker protocol would be used in cases where the client delegates the choice of discovery strategy to the OmniBroker service.

1.2. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119]

1.3. Defined Terms

[[These terms are deliberately left blank here or else we will spend time wordsmithing the defined term definitions rather than looking at the protocol.]

Authoritative DNS Server

Caching Recursive Resolver

DNS

DNS Client

Recursive Resolver

Stub Resolver

2. Architecture

PRIVATE-DNS has two parts

* Service Connection

* DNS message encapsulation

In PRIVATE-DNS, the service connection is provided by the existing [I-D.hallambaker-wsconnect] proposal. The DNS message encapsulation is new and supports encryption and authentication of the DNS protocol messages.

To make use of PRIVATE-DNS a client first establishes a connection to a DNS server (resolver or authoritative) using the connection protocol. Once a client has established a connection it MAY use it to make as many queries as desired until either the connection context expires or is cancelled by the service.

The Service Connection and Query Service MAY be operated on the same host or on separate hosts.

2.1. Service Connection

The service connection mechanism is responsible for establishing a connection context between a client and a service. The connection context comprises:

* A security context (opaque identifier, key, algorithm choice) between the client and the connection service

* One or more query host connection contexts, each comprisingNetwork connection description (IP address, Port, Protocol, transport)Security Context (opaque identifier, key, algorithm choice) between the client and the query host

The PRIVATE-DNS proposal is designed on the assumption that Service Connection transactions are relatively infrequent and thus the

efficiency of the Service Connection protocol is not a major concern.

Accordingly the Service Connection protocol is implemented as a JSON/REST Web Service over HTTP. While of an efficient encoding (e.g. [I-D.hallambaker-jsonbcd] would permit a more efficient implementation of the protocol using UDP, such an approach would be vulnerable to Denial of Service attacks against the service unless appropriate countermeasures were taken. For example use of a 'cookie' approach to prove the validity of the purported request source address.

A service connection MAY return a host connection set that includes multiple protocol and/or transport options. This has the important consequence that it allows new message formats or a transition to an entirely new protocol to be effected by simply defining a new identifier.

A distinction is drawn between a connection to a service and a connection to a host. A connection to a host is a relationship to a specific instance of a service with a distinct IP address. A connection to a service is a relationship to a set of hosts. This distinction is an important one for Denial of Service mitigation. A DNS service need not publish the same network connection description to every client. This permits a service to mitigate DoS attacks by filtering query requests by IP address, a strategy that is greatly enhanced by the large address space of IPv6.

Different configurations of the Service Connection service allow a DNS service to meet different combinations of security requirements. For example the Public Resolver described in [U-PUBLIC] would not require authentication of the client to the service but this would be required for the Subscriber, Private and Hybrid Resolvers described in [U-SUBSCRIBER], [U-PRIVATE] and [U-HYBRID].].

2.1.1. Example: Public Resolver

Following the use case [[U-PUBLIC] described in [I-D.hallambaker-dnse], Alice buys a laptop for her personal use at home. To ensure the privacy of her DNS connection she selects example.com, a public resolver that provides DNS service without requiring any form of subscription or registration.

During the initial configuration process, the machine uses the local DNS advertised in the DCHP configuration for the first and last time for discovery of the Service Connection Service of example.com.

Having discovered a Service Connection Service, the client requests a service provider for the PRIVATE-DNS service by establishing a TLS connection to indicated server. The server returns a TLS Certificate that meets the authentication criteria of the client. Once the TLS connection is established, an anonymous client connection is

established.

```
POST /.well-known/sxs-connect/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Host: localhost:8080
Content-Length: 226
Expect: 100-continue

{
  "BindRequest": {
    "Service": ["private-dns-resolver"],
    "Encryption": ["A128CBC",
      "A256CBC",
      "A128GCM",
      "A256GCM"],
    "Authentication": ["HS256",
      "HS384",
      "HS512",
      "HS256T128"]}}
```

Since the example.com service does not require authentication, the request is granted immediately and the necessary host connection parameters returned immediately:

```
HTTP/1.1 OK Success
Content-Length: 578
Date: Tue, 14 Oct 2014 19:34:07 GMT
Server: Microsoft-HTTPAPI/2.0

{
  "TicketResponse": {
    "Status": 200,
    "StatusDescription": "Success",
    "Cryptographic": [],
    "Service": [{
        "Service": "private-dns-resolver",
        "Name": "localhost",
        "Port": 9090,
        "Priority": 100,
        "Weight": 100,
        "Transport": "UDP",
        "Cryptographic": {
          "Secret": "
qJq11EcqrvWe2WfyDC2FLg",
          "Encryption": "A128CBC",
          "Authentication": "HS256T128",
          "Ticket": "
Tpau1M6HuDjwuzwLhw9SWPi9Qx1zfkcQmaj0YRnKV-JCRv2kld06zyobptvuA2F6
JGXkM0JGnSVWOPtn235wnIljsg7pZg25vPiofgPuZNY"}}]}}
```

2.1.2. Example: Hybrid Resolver

Following the use case [U-HYBRID], Alice decides to use her personal computer for work under her employer's 'Bring Your Own Device' program. Alice needs access to multiple services within her employer's intranet.

Her system administrator issues her an account name [TBS], a one time use PIN [TBS] and the DNS address of the service connection service byod.example.net. Having established a TLS connection as before, the client makes an initial request:

```
POST /.well-known/sxs-connect/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Host: localhost:8080
Content-Length: 352
Expect: 100-continue

{
  "OpenPINRequest": {
    "Service": ["private-dns-resolver"],
    "Encryption": ["A128CBC",
      "A256CBC",
      "A128GCM",
      "A256GCM"],
    "Authentication": ["HS256",
      "HS384",
      "HS512",
      "HS256T128"],
    "Account": "alice",
    "Domain": "example.com",
    "HaveDisplay": false,
    "Challenge": "
c1CfkTu5XVVLuT2gxaVFjA"}}
```

The server provides a challenge for verifying the one time use PIN.

```
HTTP/1.1 281 Pin code required
Content-Length: 511
Date: Tue, 14 Oct 2014 19:34:07 GMT
Server: Microsoft-HTTPAPI/2.0

{
  "OpenPINResponse": {
    "Status": 281,
    "StatusDescription": "Pin code required",
    "Challenge": "
9W8IxZw-bEQBbnWBWSM9Vw",
```

```
    "ChallengeResponse": "
  2FPG-xEBcYIo2137in1wxnhqUxmhygB6SsfvzhtYTXE",
```

```
       "Cryptographic": {
         "Secret": "
   KATjv8Nkix4ITrexxyGBsQ",
         "Encryption": "A128CBC",
         "Authentication": "HS256",
         "Ticket": "
   vnBXaykCug2eeRVsH-CEqhR3qJvvRQEmm4a1Ldh-G-Zqj7acqA9NtLYVCnJflaWs
   Sd2cMi8-mqdX-5VRVAMFfrxjdaQx4uq7mcr59OUFMRGSb11ZXcMkan9h142NUjmI
   t1MnYRsXWNdFndPE19zMDA"}}}
```

Having obtained the challenge value from the service, the client resends the initial request, having authenticated it this time under the challenge and one time PIN:

```
POST /.well-known/sxs-connect/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Session: Value=uuPiOYOP7kpM3xrYXMWa9JttlhR-VSf604UR6iFbPpY;
  Id=vnBXaykCug2eeRVsH-CEqhR3qJvvRQEmm4a1Ldh-G-Zqj7acqA9NtLYVCnJf
  laWsSd2cMi8-mqdX-5VRVAMFfrxjdaQx4uq7mcr59OUFMRGSb11ZXcMkan9h142
  NUjmIt1MnYRsXWNdFndPE19zMDA
Host: localhost:8080
Content-Length: 137
Expect: 100-continue

{
  "TicketRequest": {
    "Service": ["private-dns-resolver"],
    "ChallengeResponse": "
S_t81MumUqouGaxWQIT1nOJfkUaE1YcXNwQJXkXuqbM"}}
```

The server returns a set of host connections for the requested services. The scope of the PRIVATE-DNS service is limited to the domain tree *.example.net:

```
HTTP/1.1 OK Success
Content-Length: 858
Date: Tue, 14 Oct 2014 19:34:07 GMT
Server: Microsoft-HTTPAPI/2.0

{
  "TicketResponse": {
    "Status": 200,
    "StatusDescription": "Success",
    "Cryptographic": [{
        "Protocol": "sxs-connect",
        "Secret": "
UDvvBM8fE42zCs4g2mVnjw",
        "Encryption": "A128CBC",
```

```
            "Authentication": "HS256",
            "Ticket": "
```

```
WZDn4kOYJCrx6LnhuWwH3U00_aCJBcNRcUZyIV8L_hWVGjtvF8UEWTL1SgRXYcSE
zVBR9v_ER4HpSEwkYgKLX2crAo2fZMZlqyRW9kh5s88"}],
    "Service": [{
        "Service": "private-dns-resolver",
        "Name": "localhost",
        "Port": 9090,
        "Priority": 100,
        "Weight": 100,
        "Transport": "UDP",
        "Cryptographic": {
          "Secret": "
IdvuBOccKHwnPFIByHaU6w",
          "Encryption": "A128CBC",
          "Authentication": "HS256T128",
          "Ticket": "
xMVgwd-i2nHjbmZDUowVx3yAUHl_gHuh7aNzxVArYepIBMHcpaaNGw4goUsZTMby
EOUinBXDXkmVE66ExnA4H4Mgd9GSu48ReM9lKtrff98"}}]}}
```

2.2. Query Protocol Binding

The Query Protocol Binding is designed to efficiently support the following features:

* Encryption
* Prevent use in an Denial of Service attack.
* Authentication
* Multiple DNS queries and responses per PRIVATE-DNS Query [[*]
* Multiple packet responses [[*]

The features marked [[*] are not essential for the purpose of meeting the privacy requirements but considerably improve the efficiency and flexibility of the DNS protocol. In particular the ability to make multiple DNS queries in a single transaction over UDP transport enables the use of novel discovery techniques without impact on performance.

While the privacy requirements may be met through use of encryption alone, any encoding that does not provide authentication of requests allows a service to be used as an attack vector in a denial of service attack on third parties.

The Query Protocol Binding wraps the [RFC1035] message structure rather than eliminating parts that are redundant. For example, the Query Protocol Binding Transaction ID which has a minimum length of 128 bits supplements rather than replaces the DNS message transaction ID of 16 bytes.

2.2.1. Message Binding.

To ensure access to the DNS service in any network circumstance where the protocol is intentionally blocked, two message transports are specified:

UDP transport
The prefered transport providing low latency service.

HTTP Web Service

In a typical network environment where a MTU of at least 1280 bytes is supported, the UDP transport supports DNS request messages of at least 1100 bytes and responses of at least 18000 bytes.

Both transport bindings are specified in [I-D.hallambaker-wsconnect].

2.2.2. Query Protocol Example

Having established a connection to a Private-DNS service, the client from the first example performs a DNS query:

www.example.com ? A

2.2.2.1. Key Derrivation

[TBS at the moment there is no key derrivation function specified and the same key is used for encryption and authentication. This is a weak approach architecturally as a compromise of one algorithm puts the other at risk and should be fixed. Rather than use k as the key we should use MAC ("encrypt", k) and MAC ("decrypt", k) or something similar. However doing that right requires consulting past RFCs to find the right derrivation function.]

Ticket value is:

```
4e 96 ae d4 ce 87 b8 38  f0 bb 3c 0b 87 0f 52 58
f8 bd 43 1d 73 7e 47 10  99 a8 f4 61 19 ca 57 e2
42 46 fd a4 95 dd 3a cf  2a 1b a6 db ee 03 61 7a
24 65 e4 33 42 46 9d 25  56 38 fb 67 db 7e 70 9c
89 63 b2 0e e9 66 0d b9  bc f8 a8 7e 03 ee 64 d6
```

Master key is:

a8 9a b5 d4 47 2a ae f5 9e d9 67 f2 0c 2d 85 2e

Authentication key is TBS (Master)

a8 9a b5 d4 47 2a ae f5 9e d9 67 f2 0c 2d 85 2e

Encryption key is TBS (Master)

a8 9a b5 d4 47 2a ae f5 9e d9 67 f2 0c 2d 85 2e

2.2.2.2. Request

The DNS Request is: [TBS this is a placeholder]

```
;; QUESTION SECTION:
example.com.                    IN      A
```

In hex:

```
24 1a 01 00 00 01 00 00  00 00 00 00 03 77 77 77
07 65 78 61 6d 70 6c 65  03 63 6f 6d 00 00 01 00
01
```

The plaintext payload is

```
Segment(0)
Type code:              12
Segment length:         00 21
Data:
24 1a 01 00 00 01 00 00  00 00 00 00 03 77 77 77
07 65 78 61 6d 70 6c 65  03 63 6f 6d 00 00 01 00
01
```

Note that in a real world example, the request SHOULD be padded to a fixed value (e.g. 1100 bytes) to prevent traffic analysis disclosing the message contents. for illustrative purposes, a mimimal padding is applied:

The request has the transaction ID which doubles as the initialization vector of the encryption algorithm and ticket identifier prepended and the MAC value appended:

```
Transaction ID:          10 (= 16 bytes)
34 bf 46 58 50 6b 20 7a  bb 57 71 04 94 c5 80 06
Ticket:                  50 (= 80 bytes)
4e 96 ae d4 ce 87 b8 38  f0 bb 3c 0b 87 0f 52 58
f8 bd 43 1d 73 7e 47 10  99 a8 f4 61 19 ca 57 e2
42 46 fd a4 95 dd 3a cf  2a 1b a6 db ee 03 61 7a
24 65 e4 33 42 46 9d 25  56 38 fb 67 db 7e 70 9c
89 63 b2 0e e9 66 0d b9  bc f8 a8 7e 03 ee 64 d6
Encrypted Data:          00 30 (= 48 bytes)
fd e5 f6 48 69 ce 6a bb  b3 d4 ef 86 06 e9 79 f7
82 3e 86 d2 ac c5 e9 f4  b6 f3 eb a5 02 5c bf 5d
07 eb 31 cb 2b 29 90 a9  c7 96 cd bd a9 71 a1 7a
MAC:                     10 (= 16 bytes)
b9 1d e6 e4 63 93 04 d8  ff 26 8e 17 fa a9 84 aa
```

2.2.2.3. Response

The recursive resolver locates the records and returns the response.

The DNS Response is [TBS this is a placeholder]

```
;; ANSWER SECTION:
example.com.            38400   IN      A       192.168.1.20

;; AUTHORITY SECTION:
example.com.            38400   IN      NS      ns1.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.        38400   IN      A       192.168.1.28
```

In hex:

```
24 1a 81 80 00 01 00 03  00 00 00 00 03 77 77 77
06 67 6f 6f 67 6c 65 03  63 6f 6d 00 00 01 00 01
c0 0c 00 05 00 01 00 05  28 39 00 12 03 77 77 77
01 6c 06 67 6f 6f 67 6c  65 03 63 6f 6d 00 c0 2c
00 01 00 01 00 00 00 e3  00 04 42 f9 59 63 c0 2c
00 01 00 01 00 00 00 e3  00 04 42 f9 59 68
```

The plaintext payload is the DNS response plus the MAC value of the request. This response is small enough to fit into a single packet.

```
Segment(0)
Type code:               04
Segment length:          00 20
Data:
b9 1d e6 e4 63 93 04 d8  ff 26 8e 17 fa a9 84 aa
7d ff 40 00 16 91 70 d1  0a 1a 19 a5 1f 3a dc cf
Segment(1)
Type code:               12
Segment length:          00 5e
Data:
24 1a 81 80 00 01 00 03  00 00 00 00 03 77 77 77
06 67 6f 6f 67 6c 65 03  63 6f 6d 00 00 01 00 01
c0 0c 00 05 00 01 00 05  28 39 00 12 03 77 77 77
01 6c 06 67 6f 6f 67 6c  65 03 63 6f 6d 00 c0 2c
00 01 00 01 00 00 00 e3  00 04 42 f9 59 63 c0 2c
00 01 00 01 00 00 00 e3  00 04 42 f9 59 68
```

The plaintext is encrypted and the transaction identifier and MAC values added. Note that in a multiple packet response, each response has its own MAC value:

```
Transaction ID:          10 (= 16 bytes)
8e dc 41 ba 32 9f ca 6c  b4 83 43 34 88 10 7f ed
Index:                   01
Max Index:               01
Clear Response:          00 c8 (= 200)
Encrypted Data:          00 90 (= 144 bytes)
ce e6 31 cd 2d 48 01 07  23 77 db 99 ac e1 57 2a
7c f1 9b 7c cd 9e 68 8d  76 97 de 99 eb d5 bb fc
4d 17 c6 3f 9b 69 a1 e5  3a 4e 61 36 59 c6 8c 89
5c 17 2e 8c 56 6c 49 71  0a 3a 07 3d d8 1a 18 f1
25 ad 92 fa ef 85 b2 31  78 25 35 b8 e7 c2 c0 92
d3 ad a9 75 1e 10 a2 4d  3d 81 99 19 43 86 3b 29
b5 49 45 49 00 59 6b 7b  80 47 e7 fb 36 99 4b 76
45 8d aa ba e4 04 65 0b  8f 41 2e 58 df 6a ca 41
dc 16 c7 f9 ac 2a 74 ed  a4 84 80 1e e1 72 2d c9
MAC:                     10 (= 16 bytes)
49 c2 0c 8b 93 df 7f 33  4e 97 52 9a 66 2b 4f 88
```

2.2.3. Authentication Conformance

A Private-DNS server MUST authenticate queries. In the case that the UDP binding is used, a server MUST NOT make any response should the verification step fail. This requirement ensures that a Private-DNS service cannot be used to attack other systems in a Denial of Service attack through use of packets with forged source addresses.

A service MAY provide an error response in the case that a request using the Web Service binding fails as the TCP/IP connection startup provides an adequate protection against source address forgery.

2.2.4. Handling Multiple Requests

A Private-DNS service MUST accept requests that contain multiple requests. Where multiple requests are presented, each request in the transaction MUST have a unique DNS Transaction ID.

A Private-DNS service MAY limit the number of responses provided. Responses to requests MAY be returned in any order.

3. Service Connection and Key Exchangee

The Service Connection is established using [I-D.hallambaker-wsconnect]. The service identifiers for PRIVATE-DNS are as follows:

Service Identifier
: PRIVATE-DNS

Two host connection bindings are defined:

UDP Binding
: The UDP binding described in [!I-D.hallambaker-wsconnect] is the preferred host binding. The UDP binding allows most queries to be completed in a single round trip with no mandatory delays.

HTTP Binding
: A HTTP binding is specified for use as a last resort in situations where the UDP transport is not available.

3.1. UDP Binding

The prefered host connection type is to use the message encapsulation format

Protocol
: DNS

Presentation
: PRIVATE-DNS-P

Transport
: UDP

Note that the omission of version numbers in the on-the-wire data structures is intentional. Use of the message encapsulation requires that the parties have previously established a host connection

comprising the network and security parameters required to

communicate. The choice of message encapsulation including the protocol version is defined in the host connection.

In the DNS protocol requests and responses use the same message structure. The encapsulation uses different structures for requests and responses but the payload of each structure is a sequence of [RFC1035] messages.

3.2. HTTP Binding

Under certain network conditions attempts to reach the PRIVATE-DNS service may fail due to constraints imposed by firewalls or through attempted censorship. Under these conditions, HTTP [RFC2616] MAY be used as an alternative transport as follows:

Protocol
DNS

Presentation
POST

Content-Type
application/private-dns-p

Transport
HTTP

A PRIVATE-DNS service offered in this fashion MUST support HTTP/1.1 or higher. The transaction is performed as a POST request with the MIME content type application/private-dns-p.p.

4. Security Considerationsns

The broad security requirements for Private-DNS are set out in [I-D.hallambaker-dnse].

In due course this section will explain which of the security requirements is met and under which circumstances.

4.1. Confidentiality

4.2. Integrity

4.3. Access

5. IANA Considerations

6. Acnowledgements

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[I-D.hallambaker-omnibroker] Hallam-Baker, P, "OmniBroker Protocol", Internet-Draft draft-hallambaker-omnibroker-07, 21 January 2014.

[RFC2616] Fielding, R.,Gettys, J.,Mogul, J.,Frystyk, H.,Masinter, L.,Leach, P.,Berners-Lee, T., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[I-D.hallambaker-jsonbcd] Hallam-Baker, P, "Binary Encodings for JavaScript Object Notation: JSON-B, JSON-C, JSON-D", Internet-Draft draft-hallambaker-jsonbcd-01, 21 January 2014.

[I-D.hallambaker-dnse] Hallam-Baker, P, "Private-DNS", Internet-Draft draft-hallambaker-dnse-00, 21 March 2014.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, 1 November 1987.

[RFC4033] Arends, R.,Austein, R.,Larson, M.,Massey, D.,Rose, S., "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[I-D.hallambaker-wsconnect] Hallam-Baker, P, "JSON Service Connect (JCX) Protocol", Internet-Draft draft-hallambaker-wsconnect-05, 21 January 2014.

Author's Address

Phillip Hallam-Baker
Comodo Group Inc.

philliph@comodo.com