Emerging Use Cases for Encrypted DNS

Potential Next Steps for ADD

For the IETF ADD interim, January 2021

Rough consensus of: Chris Box, Tommy Jensen, Tiru Reddy, Jim Reid, Ben Schwartz
Overview

• draft-box-add-requirements-02 describes what we need to upgrade from unencrypted to encrypted, in an untrusted network.

• DEER and draft-btw-add-home suggest mechanisms for part of this space.

• What's next for ADD?

• In these slides: three scenarios beyond simple designation, with proposed requirements
An outline of the main changes vs -01

• Equivalence can mean many things, so we don't make it a requirement.

• Instead we concentrate on on the ability of an untrusted network or resolver to designate one or more resolvers.

• Designation is defined as an assertion by a network, or by a resolver, that one or more other resolvers are safe and appropriate to use without user intervention.

• Three subcases of resolver-identified: local to local, local to upstream and public to public.

• Clients still need to make their own decisions about whether and when to use designated resolvers (or not). Supplying additional information into that process would be useful.

• So we should ask ourselves which information could usefully be transported to the client to assist with that?
Three scenarios

1. DNS configuration on explicitly trusted networks
2. Resolver behavior self-description
3. Publishing and using directories of encrypted resolvers
Goals

• Solicit feedback on scope and requirements
• Gauge interest in possible next steps for the WG
• Proceed with proper requirements drafts where there is interest
• Support compliance and compatibility with other IETF standards
  • e.g. Unknown RR types, DNSSEC, Extended DNS Errors
Non-goals

• Requiring the WG to solve all three scenarios
• Taking control away from the client
• Communicating policy
  • Resolvers indicate their own behavior, not policies to impose on the client
  • Resolver selection is always left to the client
  • Policies of a managed device are controlled by the administrator
DNS configuration on explicitly trusted networks

• Client can authenticate the identity of the network (or pre-existing relationship with the network) and the user has authorized the client to override local DNS settings for a specific network.
  • BYOD devices joining Enterprise network without any MDM and configuration profile (e.g., using EAP-pwd, EAP-PSK).
  • IoT devices joining Enterprise network without a device management tool
DNS configuration on explicitly trusted networks

Goals

• Standardized discovery mechanism for BYOD and IoT devices.

• Discover local names to use Enterprise DNS server (similar to split DNS configuration in IKEv2)
  • Discover if the Enterprise network offers a split DNS configuration

• Secure Discovery of Enterprise DNS server
  • Bootstrapping Remote Secure Key Infrastructures (BRSKI) discussed in ANIMA WG for IoT devices.
  • Leverage existing secure discovery mechanisms like IKEv2 for VPN
DNS configuration on explicitly trusted networks

Non-goals

• IT-managed devices and IoT devices (using device management tool) are out of scope
• BYOD managed by MDM
• BYOD provisioned with configuration profile (e.g., Over-The-Air enrollment).
Resolver behavior self-description

Defining local-only namespaces

• Express namespaces which only this resolver can resolve
  • Authoritatively if the namespace collides with any global names
• Ex 1: Enterprise resolvers serving corporation-specific namespaces
• Ex 2: Public Wi-Fi or cellular networks offering network-local services
Resolver behavior self-description

Defining per-namespace optimization

• Express namespaces for which this resolver provides preferable resolutions
  • Ex 1: ISP routes public content requests to network caches
  • Ex 2: Public resolver designated to serve a namespace to limit parties privy to resolution
Resolver behavior self-description

Defining resolver identity

• Express information consumable by humans describing the resolver's identity

• Ex 1: Provide human-legible documentation
  • Most likely a web page link to explain server identity, terms of use, etc.
  • Not used for decision making by any protocol peer; communicated to clients for display to users

• Ex 2: Provide human-friendly description of the resolver identity
  • Friendly name and/or iconography for display in client UI identifying configured resolver
Resolver behavior self-description

Defining protocol support

• Express what optional DNS-related functionality is supported
• Ex 1: DNS Extended Errors and which codes to expect
  • Not exhaustive: server can still return any code
  • Codes 15-17 indicate kinds of filtering the resolver implements
• Ex 2: Access-controlled resolvers describing their properties outside of access control
Scenario 3: Directories of Encrypted Resolvers

Three Parties:

• Publisher: Curates a list of distinct resolvers
• Client: Fetches the list from a trusted source
• Resolvers: Identified in the list, ready for access by the client
Scenario 3: Directories of Encrypted Resolvers

Example use cases

• An application (e.g. web browser) that provides users with a list of resolvers to consider, curated by a trusted third party

• An OS vendor wants to keep its list of trusted resolvers current without requiring a software update

• A user wants to choose a resolver from a list offered by a network operator who they trust
Scenario 3: Directories of Encrypted Resolvers

Requirements

• A list can be published by a trusted network
• A list can be published at an HTTP URL
• Each resolver controls its own self-description
• Provides optional non-repudiability for the publisher
• Suitable for use in an onscreen interactive menu
• Can be used as an additional safeguard for untrusted upgrade instructions
• Uses the same protocols as the previous scenarios
Scenario 3: Directories of Encrypted Resolvers

Non-requirements

• Defending against a malicious or inept publisher
• Defending against a malicious or inept resolver
• Support for extremely long lists (e.g. >1000 resolvers)
• Combining multiple resolvers that are not sufficient independently
• Grouping related resolvers
• Enable connection without use of a bootstrap resolver
Questions?

<end of slides>