CISCO
SECURE

CISCO The bridge to possible

DNSCrypt

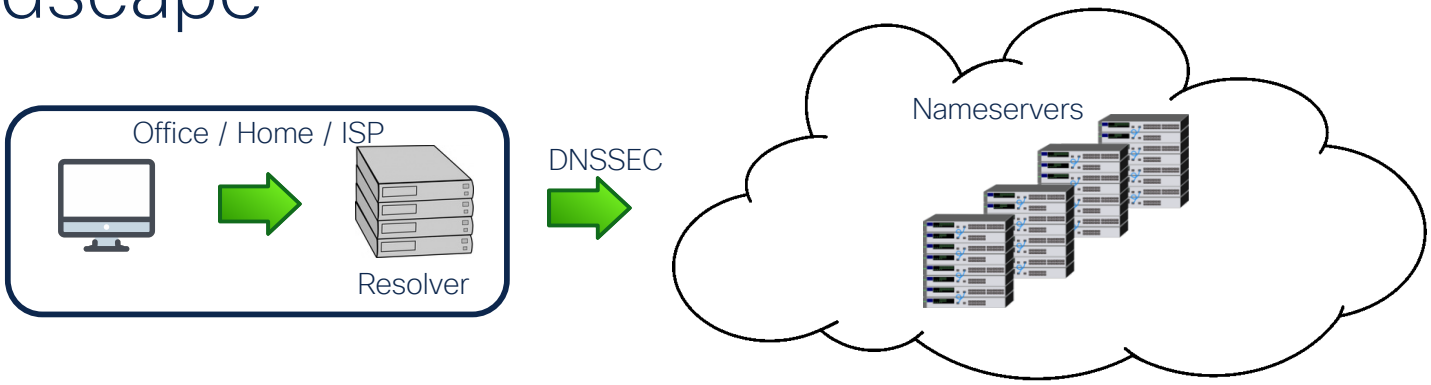
Securing Traffic from the Stub to the Resolver

Brian Somers
Principal Engineer
September 29, 2020

Agenda

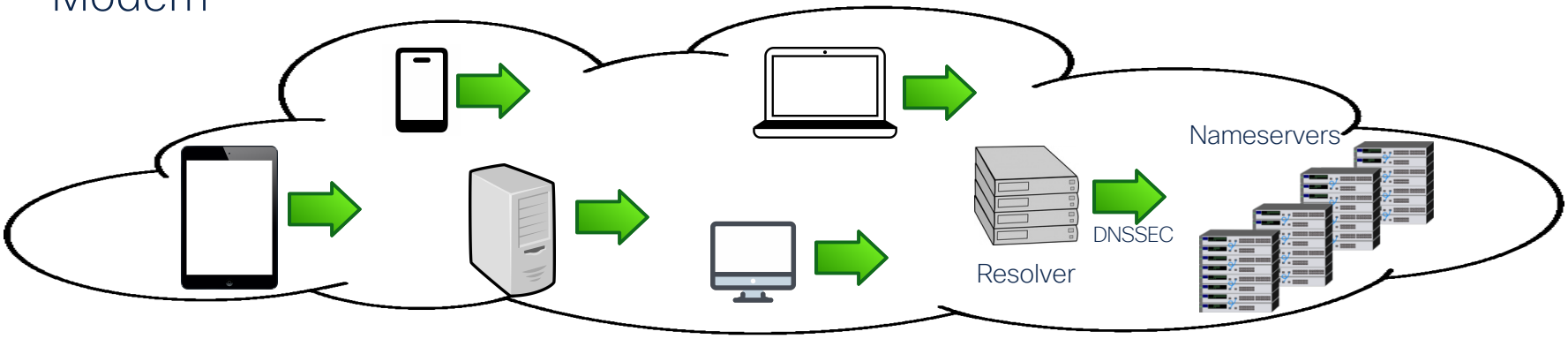
- The Landscape
- The Missing Piece
- How DNSCrypt Works
- DNSCrypt in the Wild
- TL;dr

The Landscape



Original

Modern

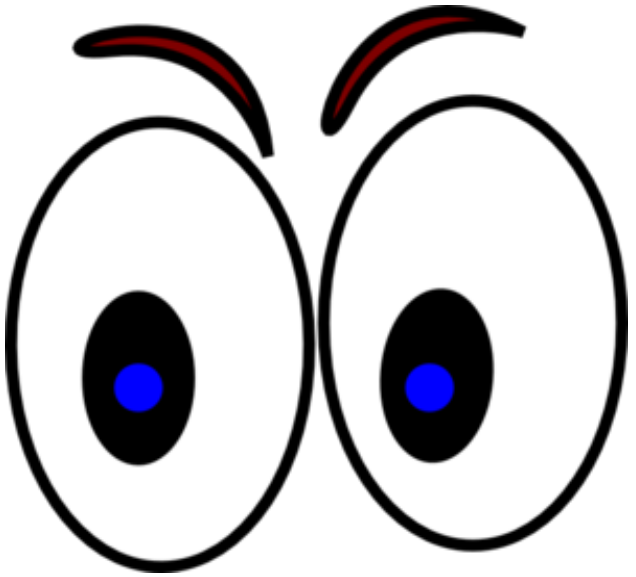


Qualities of the remote resolver

- Good caching
  High demand data will likely be cached.
  The cache is populated by huge numbers of diverse clients.
- Privacy
  Authorities learn very little about delegated lookups.
  Queries are not usually attributed to a specific client.
  Improved with qname minimization.
- Low latency
  Enterprise resolvers running on anycast addresses.
- High availability
  Massive redundancy.
  Capable of mitigating DDoS attacks.
- Security enhancements
  DNSSEC validation will ensure data integrity.
  Immediate handling of malware, phishing and inappropriate content.

The missing piece

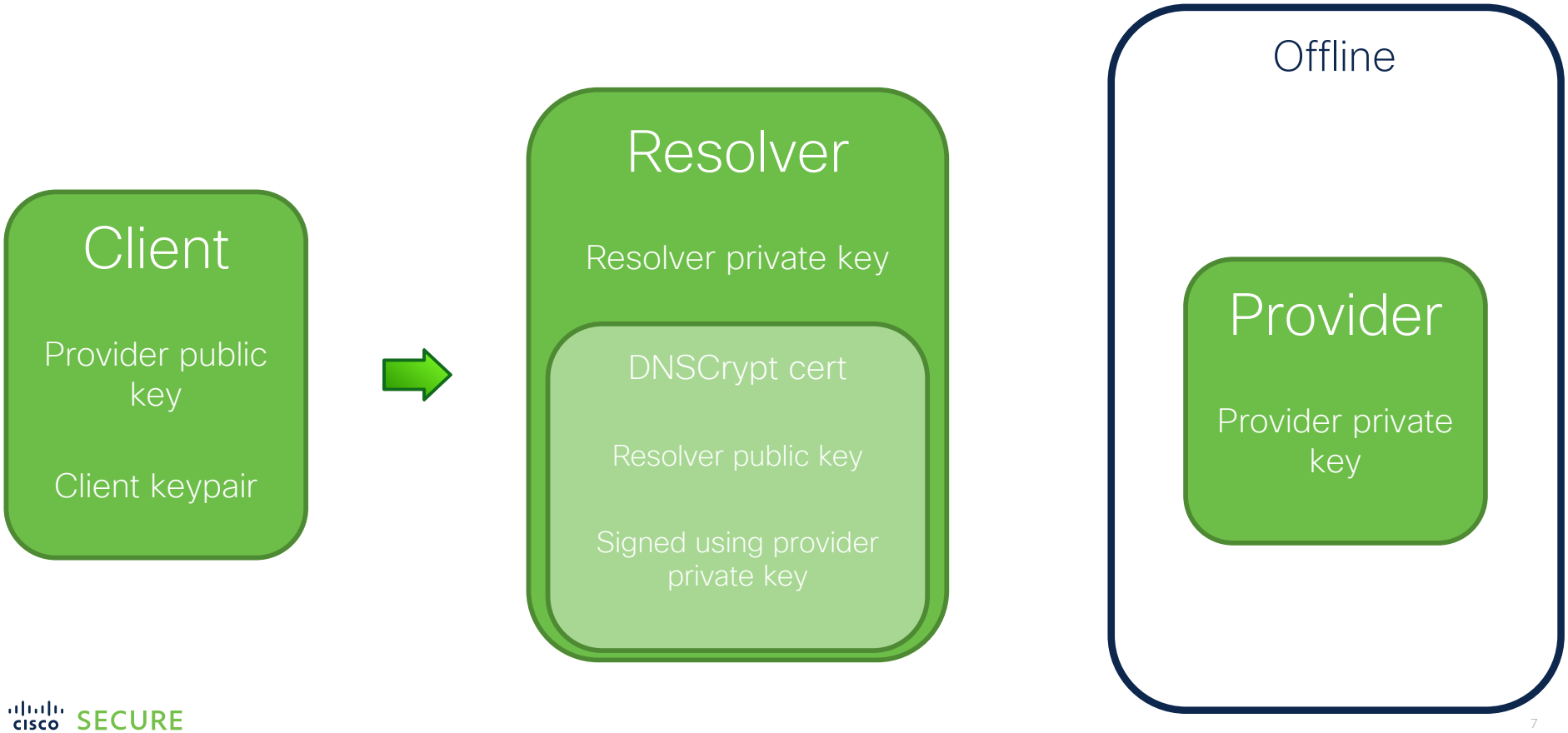
- Security from the stub to the resolver
  Without securing this data, all bets are off.
- No authentication
  The stub can't trust the AD bit.
  In fact, the stub can't trust the data!
- No privacy
  Everything is in plain sight.
  ISP eyes often mean government eyes.



DNSCrypt to the rescue

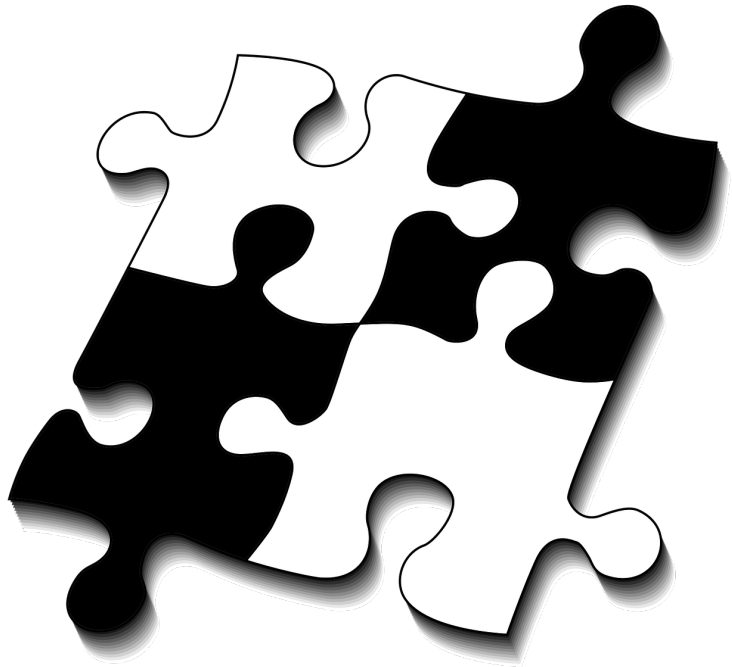
- Lives locally
  - Local network service
  - Local process
  - Built into the stub library
- Supports UDP and TCP
- 4k UDP bufsize is ok
- Uses Ec25519 – [relatively] fast
  - CPU cost is ~2.5 times
- No known vulnerabilities (since 2008)

DNSCrypt pieces



DNSCrypt movements – provider

- Infrequent (not repeated for many many years)
  Provider creates keypair, hides the private key.
  DNSCrypt client software is configured with the public key.
- Occasionally (one or more times per year)
  Provider generates a new resolver keypair.
  Provider installs the resolver private key on the resolver.
  Provider creates a DNSCrypt certificate.
  - Embeds the resolver public key
  - Signs using the provider private key
  - Publishes as a DNS TXT RR

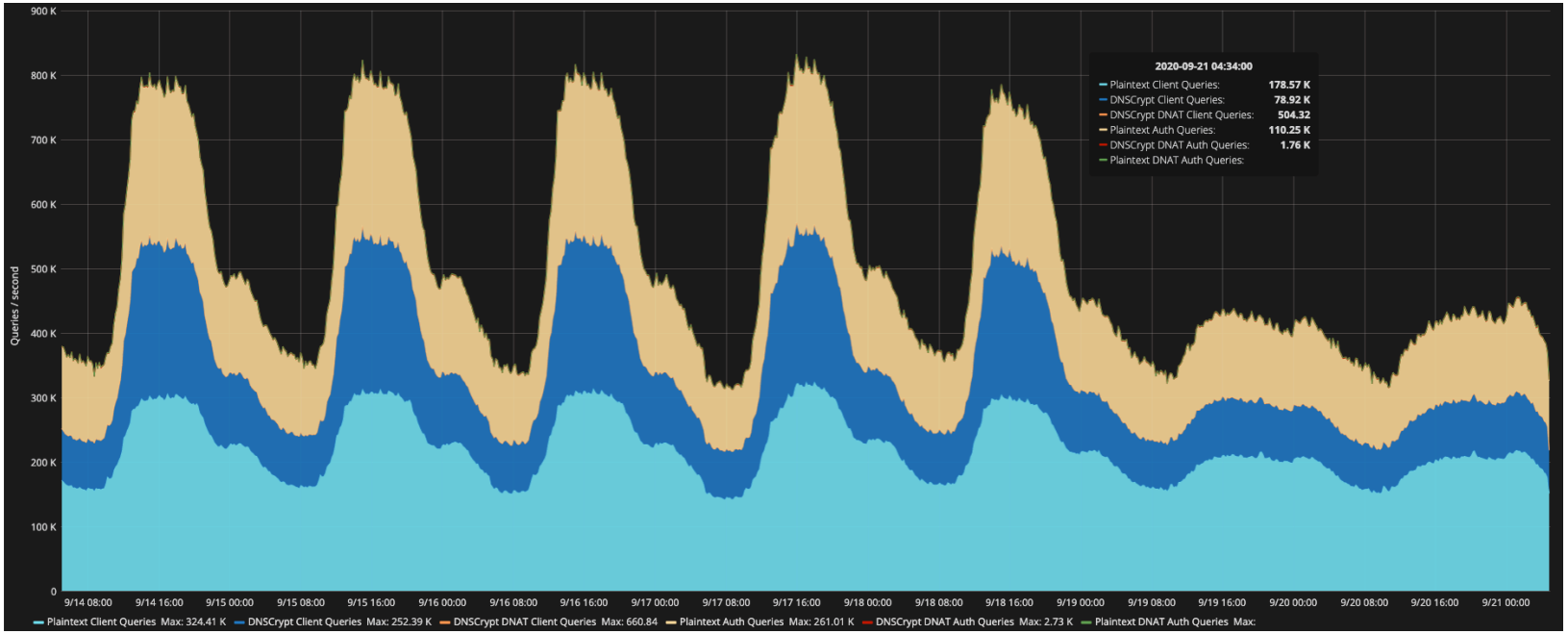


DNSCrypt movements – client/resolver

- Every hour
  Client refreshes the DNSCrypt certificate TXT RR in its cache.
  Client validates the DNSCrypt certificate using the provider public key.
  Client chooses its favourite key.
- Periodically (per session or per query)
  Client generates a client keypair.

- For each query
  Client encrypts using the resolver public key from the DNSCrypt certificate.
  Client embeds its public key in the query.
  Resolver decrypts the query using the resolver private key.
  Resolver encrypts the response using the client public key.
  Client decrypts the response using the client private key.
- https://dnscrypt.info/protocol

DNSCrypt in the wild

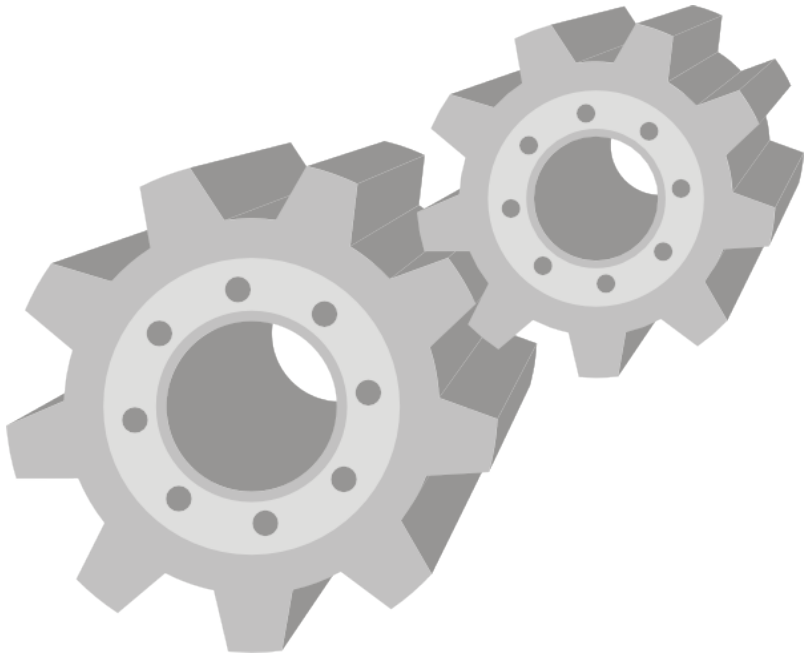


1 week

- Client (light blue), DNSCrypt (dark blue), Authoritative (yellow)

DNSCrypt TL;dr

- Privacy
  Captured traffic cannot be interpreted.
- Authentication
  The AD bit can be trusted.
- Administrative sanity
  No increased packet counts (although packets **are** larger).
  Traffic patterns are the same.
  Traffic is identifiable (queries and responses have **magic**).



Next steps

- Many implementations
  https://dnscrypt.info/implementations
  Some support DNSCrypt, DoH and DoT – comparisons are easy.
- Many services
  https://dnscrypt.info/public-servers
- Frequently Asked Questions
  https://dnscrypt.info/faq
- A DNSCrypt RFC
  Doesn't yet exist
  It's probably well overdue!