Private Discovery with TLS and ESNI

Christian Huitema
draft-huitema-dnssd-tls-privacy-00
Assumptions

• TLS 1.3
  • ClientHello +SNI ➔
  • ← ServerHello + encrypted extensions (encrypted Cert)

• Encrypted SNI
  • Client-facing Server publishes ESNI public key
  • SNI replaced by ESNI, encrypted with public key
  • Only intended server can decrypt ESNI, forward to designated SNI

• UDP transport based on TLS 1.3
  • DTLS or QUIC
Basic Idea: Multicast ESNI

• Learn SNI encryption key of designated server,
• Broadcast / Multicast first UDP Packet,
  • Includes TLS 1.3 ClientHello + ESNI
• Servers listen to multicast requests
  • Trial decryption of ESNI
  • If decrypted & matches local value, send back unicast response
• Establish 1-1 connection
• Maintain TLS and ESNI guarantees of security, and privacy
Rely on Secret Discovery Key

• Standard ESNI publishes ESNI public key in DNS
  • Would allow anyone to discover whether server is on-line

• Fix: provision ESNI public key only to authorized clients
  • Rename “ESNI public key” as Discovery Key, meant to be kept secret
  • Only the server knows the private key

• Result: resilience
  • If discovery key compromised, server can be discovered but clients remain private
Optional two-phase model

• What if server is not using DTLS or QUIC?
• Fix: discover DNS server
  • Use DNS over DTLS or DNS over QUIC
  • Discover private DNS server using private discovery (TLS, ESNI)
  • Private DNS transactions to get DNSSSD data for the server
Remaining gap, scaling issues

• Too many messages?
  • Client sends one discovery broadcast per target server
  • But in “application level” scenarios, given client is interested in few servers

• Polling for servers?
  • If server not present when query is sent, client will have to retry
  • But in “peer to peer” scenario, roles are symmetric, peer polls on arrival

• Possible fix for next version: server announce
  • Add “I am here” message, verifiable with server discovery key
  • Every client tries verifying with every known discovery key
  • If client detects server, establishes connection immediately
Remaining gap, forward privacy

• If public discovery key is compromised:
  • Server can be discovered or tracked using active queries
  • Clients remain private

• If private discovery key is compromised
  • Server can be spoofed
  • Old logs can be analyzed

• Possible fix: frequent key rotations
  • Will require provisioning mechanism
Next steps?