Using Confidential Computing to protect DNS resolution

draft-arkko-dns-confidential-01.txt

Jari Arkko & Jiri Novotny
Ericsson Research

General Context

DNS privacy is a popular topic!

- Domain name meta-data visible on the wire (even with encryption)
- Resolvers have the potential too see user’s entire browsing history
- Large resolver services are an attractive target (public, operator, …)

⇒ Work on encrypted DNS query protocols (e.g., DoT, DoH, DoQ)
⇒ Discovery mechanisms
⇒ Practices, expectations, contracts (e.g., RFC 8932, Mozilla’s TRR reqs)
⇒ Improvements outside DNS (e.g., eSNI)
⇒ General technology developments (e.g., confidential computing, RATS)

Context of This Work

Build on all that work – and assume all communications encrypted

- What problems remain? Is there a next step?

- Our worry is that resolvers can be a major remaining source of leaks
  - Accidents, attacks, commercial use, authorities request
- We need to protect user's data in flight, at rest, or in use – we wanted to experiment with tech that could reduce leaks on the last two
- There are some possibilities, but we wanted to talk about it to you and get your feedback – there are operational impacts

DNS Resolvers with Confidential Computing

Basic idea: run the resolver process inside a TEE, and do not provide any user-specific data outside the encrypted process

- Even to the DNS operator itself
- Or the owner of computer hardware, or the operating system

Requires a willing DNS operator to do this, of course, and may require trusting a third party in some cases.

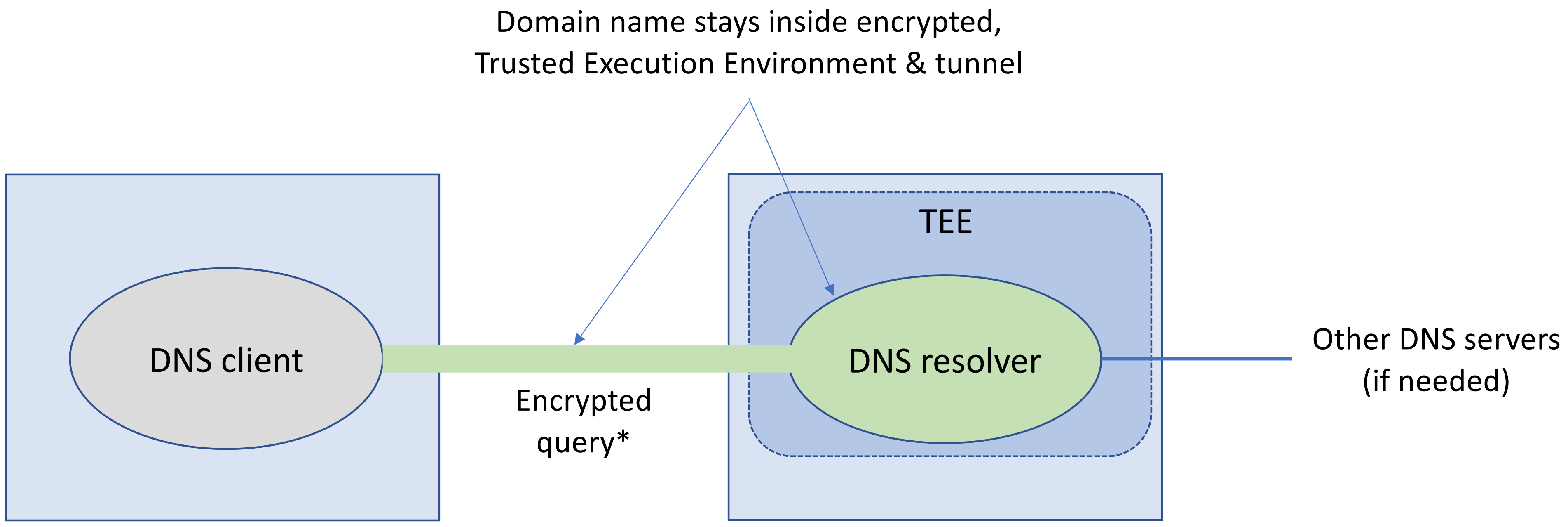
Trusted Execution Environment (TEE)

”An environment that enforces that any code within that environment cannot be tampered with, and that any data used by such code cannot be read or tampered with by any code outside that environment." [ietf-teep-architecture]

- Data confidentiality and integrity
- Code integrity
- Attestability

Hardware-based TEEs from Intel SGX, ARM Trustzone etc.

DNS Resolvers with Confidential Computing


Domain name stays inside encrypted,
Trusted Execution Environment & tunnel
DNS client
TEE
DNS resolver
Encrypted
query*
Other DNS servers
(if needed)

* Can be built on top of DoT [RFC7858] or DoH [RFC8484]

Considerations (1)

- Upside: It is possible to hide all user-related information
  - We argue that we should do so
- Upside: Can provide evidence of compliance to an expected practice
  - “Trust but verify”
- Downside: Changes operational practices
  - Scaling, systems monitoring, debugging
  - Denial-of-service
  - Can be addressed (we think) at least to an extent
    - Can’t reveal individual user behaviour
    - Can reveal aggregate statistics on load, performance, memory, cache, errors, etc.
    - Limits on particular types of requests or from particular clients

Considerations (2)

- Upside: May change business practices, e.g., on use of data
    - Could still provide aggregate data, e.g., most popular lists, etc.
    - Can still perform geolocation etc (inside TEE)
- Downside: Dependencies
    - Suitable server hardware
    - Use of attestation leads to the need to have a trusted party
    - If clients check attestation this requires changes
    - Who checks the software?
- Observation: Not a cure for all problems, and attacks remain
    - If you thought $MANUFACTURER CPUs were secure, we have news for you
    - Also, side channel attacks
    - Or software issues
    - May still provide an additional layer of defense
    - Also, this is an additional hurdle for an attacker to have to overcome, e.g., $AGENCY

Conclusion

- Data held by servers SHOULD receive at least as much security attention as communications do.

- Particularly crucial for DNS (potential leak of user's browsing history), but principles apply also to other services.

- Consider all applicable tools – one discussed here
- Consider the operational and business implications of such tools
- Feedback to us very welcome – we are here to hear your thoughts
  - Feasible or science fiction?
  - Need to take X, Y, Z into account?

Thank you!