Trabalho 1 – Resolver Recursivo Validante com Cache e DNSSEC (23set2025)

Redes de Computadores 2025/2 - Prof. Irineu Sotoma

Objetivo Geral

Desenvolver, em qualquer linguagem de programação (exceto Python), um resolver DNS recursivo validante, com suporte a cache (via servidor de cache) e DNSSEC. O resolver deve utilizar apenas chamadas de sockets de baixo nível (socket, bind, connect, accept, send, recv, sendto, recvfrom, close, shutdown) para enviar e receber mensagens.

Requisitos Funcionais

1. Resolução recursiva mesmo quando servidor de nomes é iterativo (RD=0), com suporte a:

- Delegações (NS + glue records).
- CNAME encadeados.
- Respostas negativas (NXDOMAIN, NODATA).
- Fallback para TCP quando TC=1. Consultas DNS normalmente usam **UDP/53**. Porém, se a resposta for muito grande (ex.: muitas chaves DNSSEC, registros longos), o servidor pode truncar a resposta. Isso é sinalizado pelo **bit TC=1 (Truncated)** no cabeçalho DNS. Quando isso ocorre, o resolver deve **refazer a mesma consulta em TCP/53**, que permite mensagens maiores e completas.

2. Suporte a DNS over TLS (DoT):

- Porta 853.
- Handshake TLS e SNI (Server Name Indication: campo no handshake onde o cliente informa **qual hostname** está tentando acessar, para receber o certificado certo).
- Validação de certificado.

3. DNSSEC:

- Carregar âncoras de confiança (DNSKEY). Uma **âncora de confiança** é uma chave pública que o resolvedor já conhece (ex.: a **KSK do root**). Elas são armazenadas em arquivo (`root.keys`) e usadas como ponto de partida para verificar assinaturas.
- Validar cadeia DS → DNSKEY → RRSIG: **DS (Delegation Signer)**: no pai, guarda o hash da DNSKEY do filho. **DNSKEY**: chave pública da zona (no filho). **RRSIG**: assinatura que cobre um conjunto de registros (RRset). Processo: 1) Resolver pega o **DS** da zona pai. 2) Compara se bate com o hash de alguma **DNSKEY** publicada pelo filho. 3) Usa a **DNSKEY** correspondente para verificar a **RRSIG** que cobre os registros recebidos. 4) Assim garante que a resposta é autêntica.
- Verificar validade temporal do RRSIG: Todo RRSIG traz: **Inception**: quando a assinatura começa a valer. **Expiration**: quando expira. O resolver deve rejeitar assinaturas fora dessa janela de tempo.
- Suportar RSA/SHA-256 (8) e ECDSA P-256/SHA-256 (13): há bibliotecas em várias linguagens de programação.
- Marcar AD=1 se validado, AD=0 caso contrário.

4. Multithread:

- Consultar múltiplos NS em paralelo (fan-out).
- Pool de threads para consultas concorrentes.
- Cache thread-safe.

5. Cache:

- Respeitar TTL em respostas positivas (registros válidos), RFCs 1034 e 1035.
- Negative caching (RFC 2308): armazena respostas negativas (NXDOMAIN (o nome não existe em nenhuma zona) ou NODATA (o nome existe, mas não há registro do tipo solicitado.)).
- Expurgar itens expirados.

- O servidor de cache será um processo daemon, cujas caches positiva e negativa estarão totalmente em memória principal.

- Se o daemon não estiver ativo, o resolver deverá fazer a consulta normalmente.

- Tentar ativar novamente, se já estiver ativo, irá somente gerar mensagem de erro.

- O tamanho padrão das caches positiva e negativa é de 50 entradas cada.

- Parâmetros do daemon: --activate (ativa o servidor de cache), --deactivate (expurga todos os itens da cache e mata o daemon) , --status (Informa o tamanho máximo das caches positiva e negativa, e quantidade de entradas efetivamente utilizadas em cada uma), --set positive (define uma nova quantidade de entradas da cache positiva), --set negative (define uma nova quantidade de entradas da cache negativa), --purge positive (expurga a cache positiva), --purge negative (expurga a cache negativa), --purge all (expurga todos os itens da cache), --list positive (lista os itens da cache positiva), --list negative (lista os itens da cache negativa), --list all (lista os itens das caches positiva e negativa).

Requisitos Técnicos

Implementação em C, C++, Java, Scala, Go, etc. (exceto Python).

Uso apenas de sockets básicos para comunicação TCP/UDP.

Não utilizar soluções prontas de DNS.

Interface em linha de comando do resolver com parâmetros como --ns, --name, --qtype, --mode, --sni, --trust-anchor, --fanout, --workers, --timeout, --trace.

Registros DNS – Enviados e Recebidos

Exemplos de Registros enviados (consulta)

Header: ID, flags, QDCOUNT.

Question Section: QNAME, QTYPE, QCLASS.

Exemplos de Registros recebidos (resposta)

Header: ID, flags, QDCOUNT, ANCOUNT, NSCOUNT, ARCOUNT.

Answer Section: resposta direta à consulta (A, AAAA, CNAME, RRSIG).

Authority Section: dados de delegação (NS, SOA, DS).

Additional Section: glue records (A/AAAA dos NS da própria zona) e registro OPT (EDNS).

Trabalho 1 – Resolver Recursivo Validante com Cache e DNSSEC (23set2025)

Redes de Computadores 2025/2 - Prof. Irineu Sotoma

Resumo dos tipos de registros (podem ser necessários outros)

Tipo	Onde aparece	Descrição
A	Answer / Additional	Endereço IPv4
AAAA	Answer / Additional	Endereço IPv6
ANCOUNT	Header	Número de registros na seção **Answer**.
ARCOUNT	Header	Número de registros na seção **Additional**.
CNAME	Answer	Alias para outro nome
DNSKEY	Answer	Chaves públicas da zona (DNSSEC)
DS	Authority	Hash de DNSKEY do filho (DNSSEC)
Flags	Header	Bits de controle: QR, Opcode, AA, TC, RD, RA, Z, AD, CD, RCODE.
ID	Header	Identificador único da consulta/resposta.
MX	Answer	Mail Exchanger (servidor de e-mail do domínio).
NODATA	Answer	O domínio existe, mas não há registros do tipo solicitado.
NS	Authority	Servidores autoritativos de uma zona
NSCOUNT	Header	Número de registros na seção **Authority**.
NXDOMAIN	Answer (RCODE=3)	Erro: o nome de domínio não existe.
OPT	Additional	Registro especial de EDNS (Extension Mechanisms for DNS, RFC 6891). Essencial para DNSSEC, porque assinaturas e chaves tornam as respostas muito maiores. DNS clássico (sem EDNS) só garante até **512 bytes** em UDP. Com OPT, cliente e servidor podem negociar **até 4096 bytes** (ou mais).
QCLASS	Question	Classe de rede (IN para Internet).
QDCOUNT	Header	Número de registros na seção **Question**.
QNAME	Question	Nome de domínio consultado.
QTYPE	Question	Tipo de registro solicitado (A, AAAA, MX, etc.).
RRSIG	Answer / Authority	Assinatura digital de um RRset (DNSSEC)
SOA	Authority	Dados da zona (autoridade, serial, timers)

Trabalho 1 – Resolver Recursivo Validante com Cache e DNSSEC (23set2025)

Redes de Computadores 2025/2 - Prof. Irineu Sotoma

Critérios de Avaliação

Resolução recursiva, e também ao acessar servidores iterativos (DNS e DoT) ..... 4,0 pontos

Cache positivo e negativo com servidor de cache ..... 3,0 pontos

Qualidade do código (documentação (código e README) e arquitetura) ..... 1,5 ponto

Relatório ..... 1,5 ponto

Resolução recursiva, e também ao acessar servidores iterativos com Validação DNSSEC (DS, DNSKEY, RRSIG, AD) ..... 1,0 ponto (Bônus)

Multithread (fan-out + pool) ..... 1,0 ponto (Bônus)

Bônus de 1,0 ponto para os grupos formados por 2 ou 3 integrantes, e definidos até 15/09/2025.

Penalidade de 5,0 pontos se houver implementação de conexões via socket (UDP/TCP/DoT) que não seja via chamadas de sockets de baixo nível.

A utilização de soluções prontas para DNS, ou detecção de plágio, irá levar à nota ZERO no Trabalho Prático TP1.

Entregáveis (Via AVA)

1. Código-fonte completo.

2. README com instruções de compilação/execução do resolver, e com descrição sucinta de utilização do servidor de cache.

3. Relatório (4–7 páginas), em PDF, explicando arquitetura, decisões e testes.

4. Arquivo de âncoras (root.keys).

Prazo

Entrega até 15 de outubro. Grupos de até 3 estudantes.

Explicação dos parâmetros de linha de comando:

--ns: servidor de nomes a ser consultado (IP ou domínio).
--name: nome de domínio alvo da consulta (ex.: www.exemplo.com).
--qtype: tipo de registro solicitado (A, AAAA, MX, NS, etc.).
--mode: modo de operação do resolver (recursive, iterative, validating, etc.).
--sni: Server Name Indication, usado em conexões TLS para indicar o hostname.
--trust-anchor: arquivo de âncoras de confiança (ex.: root.keys) para validação DNSSEC.
--fanout: número máximo de consultas simultâneas a servidores NS (paralelismo).
--workers: tamanho do pool de threads para execução concorrente.
--timeout: tempo máximo de espera por resposta antes de considerar falha.
--trace: ativa rastreamento detalhado da resolução, exibindo cada passo do processo.

Impacto dos parâmetros de linha de comando nas funcionalidades do Trabalho 1:

--ns → Resolução recursiva e fallback TCP/DoT; define o servidor inicial a ser consultado.

--name → Resolução recursiva (delegações, CNAME); cache positivo/negativo; nome alvo da consulta.

--qtype → Tratamento de respostas positivas e negativas (NXDOMAIN, NODATA); tipo de registro solicitado.

--mode → Define o modo do resolver (recursivo, iterativo, validating); impacta cache e DNSSEC. Os vários modos auxiliarão na depuração do código.

--sni → DNS over TLS (DoT); garante apresentação correta do certificado.

--trust-anchor → Validação DNSSEC (DS, DNSKEY, RRSIG, AD); especifica âncora de confiança.

--fanout → Multithread – consultas paralelas a múltiplos NS; acelera resolução.

--workers → Multithread – tamanho do pool de threads; impacta concorrência e escalabilidade.

--timeout → Resolução recursiva e fallback; tempo máximo de espera por resposta antes de trocar de servidor.

--trace → Depuração e validação; exibe passo a passo da resolução (delegações, cache, DNSSEC).

Modos de operação (--mode) do resolver e seus impactos:

recursive
Flags/Configuração: RD=1
Impacto: Executa resolução completa a partir do root; usa cache; faz fallback TCP; pode validar DNSSEC se usado com --trust-anchor.

forwarder
Flags/Configuração: RD=1
Impacto: Encaminha consultas para outro servidor (definido em --ns); simplifica lógica mas mantém cache local.

iterative
Flags/Configuração: RD=0
Impacto: Consulta iterativa; mostra Authority e Additional; útil para depuração e análise de delegações (NS + glue records).

validating
Flags/Configuração: RD=1 + validação DNSSEC
Impacto: Verifica cadeia DS→DNSKEY→RRSIG; marca AD=1 em respostas válidas; rejeita inválidas.

insecure
Flags/Configuração: RD=1, ignora DNSSEC
Impacto: Não valida assinaturas; AD=0; permite comparar desempenho e depuração em falhas de DNSSEC.

strict-dnssec
Flags/Configuração: RD=1 + validação obrigatória
Impacto: Aceita apenas respostas DNSSEC assinadas; rejeita zonas sem suporte a DNSSEC.

dot
Flags/Configuração: RD=1 + TLS/853 + SNI
Impacto: Consulta usando DNS over TLS; requer handshake TLS; permite comparar segurança e latência com UDP/TCP.