

SVCB, HTTPS, DDR a další novinky v DNS

Petr Krčmář



2. listopadu 2024



Uvedené dílo (s výjimkou obrázků) podléhá licenci Creative Commons Uveďte autora 3.0 Česko.

- linuxák od roku 1998
- správce serverů
- lektor a konzultant
- šéfredaktor [Root.cz](#)
- člen [vpsFree.cz](#)
- organizátor [LinuxDays](#)
- můj web je [petrkrcmar.cz](#)



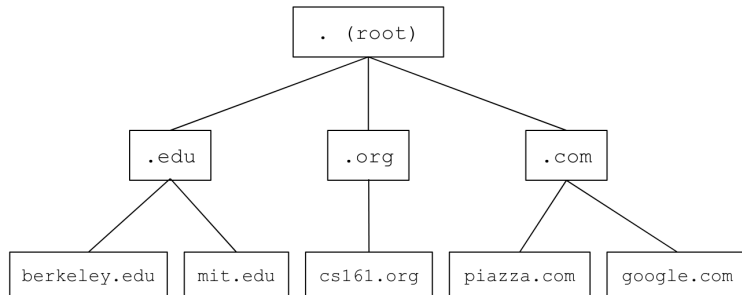
<https://www.petrkrcmar.cz>

Princip DNS

Pár slov o DNS

- informace jsou v DNS rozloženy v doménovém stromě
- celé jméno se skládá z domén oddělených tečkami
 - ananas.do.petrkrcmar.cz
 - vlevo konkrétní, vpravo nejobecnější
- popisuje také **průchod** doménovým stromem
 - podle něj postupujeme při hledání odpovědi
- nejprve je třeba jít ke kořeni a zjistit informace o .cz
- poté podle nich postupovat do dalšího patra
- nakonec identifikujeme konkrétní **uzel** ve stromě
- hloubka omezena na 127 úrovní
 - obvykle využíváme maximálně čtyři až pět

Stromová struktura



Autoritativní server

- běžný počítač vystupující na internetu
- je na něm spuštěný DNS démon (služba)
- drží skutečná data, která nás zajímají
 - server odpovídá na DNS dotazy klientů
- je zodpovědný jen za určitou oblast (zónu)
 - nemusí tedy držet data pro celý internet
- u určitých dat může **odkazovat** na jiný server
 - přesně takto je organizován doménový strom

- jednotka informace uložená ve stromu DNS
- uložena jako zdrojový záznam (resource records, RR)
- pro každé jméno **libovolný počet**, i **opakujících** se typů
- každý záznam obsahuje vždy pět položek

jméno (name) ke kterému se vztahuje

životnost (time to live), TTL v sekundách (až 68 let)

třída (class) vždy hodnota 1, internet (IN)

typ (type) typ předávané informace (16 bitů)

data samotná informace, formát závisí na typu

Běžné typy DNS záznamů

SOA základní informace o doméně

A adresa protokolu IPv4

AAAA adresa protokolu IPv6

CNAME přezdívka jiného jména

PTR jméno k IP adrese

NS jméno DNS serveru pro doménu

MX jméno poštovního serveru pro doménu

TXT další informace v textové podobě

DNS pro web

Tradiční použití na webu

- klient (prohlížeč) chce načíst obsah
- musí zjistit informace o doméně z URL
- zeptá se tedy na **záznamy pro dané jméno**
 - ptá se na A záznam pro IPv4
 - ptá se na AAAA záznam pro IPv6
 - dotazy jsou asynchronní (Happy Eyeballs)
- jakmile dostane odpověď, spojuje se k serveru
 - naváže spojení TCP na výchozí port
 - poté se ptá pomocí HTTP
 - volitelně ještě mezivrstva TLS

Proč to nestačí?

- máme **pevnou vazbu** mezi identitou a umístěním serveru
- nedokážeme signalizovat **port**, na kterém služba běží
 - používáme výchozí port nebo ho musíme předat prohlížeči
- nedokážeme signalizovat **protokol** aplikační vrstvy
 - přechod na HTTP/2 nebo HTTP/3 vyžaduje další komunikaci
- nedokážeme signalizovat **bezpečný transport**
 - třeba aby se rovnou předsadil TLS před HTTP
- nedokážeme použít **alias** celé domény
 - záznam CNAME nesmí být v apexu zóny (kombinace se SOA a NS)
- přenášíme tedy **pouze IP adresy** – víc neumíme

Záznam typu SRV

- některé služby umějí záznam typu SRV (service), typ 33
- definován v [RFC 2782](#)
- **upřesňuje** informace o serveru poskytujícím službu
 - jméno hostitele, transportní protokol, číslo portu, prioritu...
- použití vždy vyžaduje **další dotaz** do DNS pro zjištění IP
- používají ho klienti LDAP, SIP, XMPP...
 - na webu **není podporován**
- elektronická pošta má podobný vlastní záznam typu MX

Příklad záznamu SRV

```
_sip._tcp.example.com. 86400 IN SRV 0 5 5060 sipserver.example.com.
```

Proč nemáme na webu SRV?

- protože to RFC zakazuje
 - když s tím nepočítá protokol, klient nesmí používat
 - viz *Applicability Statement*
- bylo několik pokusů to vyřešit (2002, 2009, 2014...)
 - vždy skončilo jen u návrhů a nedořešených problémů
- obvykle se zmiňuje **zvýšení latence** a **složitost integrace**
 - některé nedořešené problémy mají bezpečnostní dopady
- SRV není příliš vhodný pro dnešní web
 - je **nerozšiřitelný** a má pevný formát
 - je obvykle **povinný**, což komplikuje jeho doplnění do standardu
- tvůrci prohlížečů ho proto nechtějí a nepodporují

Nové záznamy SVCB a HTTPS

- definovány v [RFC 9460](#) z listopadu 2023
 - HTTPS je zvláštní případ obecnějšího SVCB (SerViCe Binding)
- poskytují **všechny informace** pro připojení ke službě
 - lepší výkon a soukromí, netřeba vyjednávat změny
 - jsou **rozšiřitelné** – můžeme je dále doplňovat
- všechno v **jednom** DNS záznamu
 - méně dotazů znamená rychlejší spojení
 - rovnou se spojím správně a se správnou službou
- odděluje se identita (doména) od hostitele služby
- nová metoda doménového aliasu
 - pro weby na `https://example.com` si zajdi k `svc.example.net`

DNS záznam HTTPS

Záznam typu HTTPS

- DNS záznam typu 65 (RR type)
 - pokud vaše nástroje tento typ záznamu ještě neznají
- záznam obsahuje prioritu, podobně jako MX
 - záznam s nejnižší prioritou se použije jako první
- následuje seznam parametrů ve formátu **klíč=hodnota**
 - parametry oddělujeme mezerou, více hodnot oddělujeme čárkou

Dotaz na záznam HTTPS

```
$ dig HTTPS example.com
$ dig TYPE65 example.com
example.com 3600 IN HTTPS 1 . alpn="h3,h2" ipv4hint="192.0.2.1" ipv6hint="2001:db8::1"
```

Použitelné parametry

- `alpn` protokol podporovaný serverem
- `no-default-alpn` ignoruj výchozí protokol
- `port` TCP nebo UDP port služby
- `mandatory` položky nutné k navázání spojení
- `ipv4hint` seznam IPv4 adres
- `ipv6hint` seznam IPv6 adres
- `ech` encrypted ClientHello
- `dohpath` cesta pro poskytování DoH
- `ohhttp` parametry pro HTTP/3

Seznam voleb je v [registru v IANA](#).

ALPN (Application-Layer Protocol Negotiation)

- ALPN byl původně vyvinut pro TLS ([RFC 7301](#))
 - rozšiřující hlavička pro vyjednání protokolu
 - posílá se rovnou v ClientHello a oznamuje *Next Protocol*
- v současnosti jsou použitelné volby:
 - HTTP/1.1 (http/1.1)
 - HTTP/2 over TLS (h2)
 - HTTP/2 over TCP (h2c) – nešifrovaná
 - HTTP/3 (h3)
- pořadí voleb je důležité, ukazuje **preferenci** – první je nejsilnější
- volba `no-default-alpn` zakazuje klientovi použít výchozí protokol
 - klient pak musí použít některou z hodnot navržených serverem

IPv4 a IPv6 adresy

- položky `ipv4hint` a `ipv6hint` obsahují seznam IP adres
- tyto adresy by měly být použity, pokud klient ještě žádné nezná
 - používá se ke zrychlení komunikace, klient má informace dřív
- pokud klient už IP adresy odněkud zná, měl by nápovědy **ignorovat**
- adresy mohou být použity zcela asynchronně
 - klient si vybírá podle vlastních algoritmů
 - může použít metody jako Happy Eyeballs
- klient může stále využít klasické záznamy A a AAAA
 - pokud něco selže nebo jsou adresy z nápovědy nedostupné

- položka port umožňuje klienta poslat na **nestandardní port**
 - to pomocí klasického záznamu A/AAAA není možné
- jde o číslo portu pro TCP nebo UDP
 - záleží na zvoleném transportním protokolu
- musí jít o jediné číslo v rozsahu 0 až 65535
 - více položek není povoleno
 - stejně jako jiné znaky v hodnotě

Encrypted ClientHello

- položka ech (dříve ESNI) zlepšuje bezpečnost a soukromí
- dovoluje šifrovat už úvodní paket TLS (ClientHello)
 - jde o rozšíření v TLS 1.3
 - úvodní seznámení má **vnější** a **vnitřní** část
 - ta vnější je už v DNS, vnitřní pak jde standardní cestou
- zabrání se přenosu SNI v otevřené podobě
 - kvůli výběru certifikátu klient prozrazuje doménu
- ECH musí podporovat klient i server
 - u serveru to zahrnuje šifrovací knihovnu (OpenSSL)
- záznam obsahuje **veřejný klíč** a podrobnosti k protokolu
- v serverech je zatím experimentální podpora
 - v distribucích ji nehledejte, musíte si sestavit
 - doplnit do webového serveru i šifrovací knihovny

Režim alias

- zvláštní režim, kdy je použita **priorita 0**
 - pozor na to při číslování priorit v běžném režimu
- obvykle se používá jeden takový aliasový záznam
 - pokud je jich víc, klient zvolí jeden náhodně
- používá se v apexu zóny, kde nelze použít CNAME
 - na rozdíl od CNAME zůstávají v platnosti ostatní záznamy
 - alias se týká vždy jen jedné dané služby

Příklad aliasu

```
petrkrccmar.cz.      86400      IN      HTTPS   0 www.petrkrccmar.cz.
```

DNS záznam SVCB

Záznam typu SVCB

- DNS záznam typu 64 (RR type)
- zobecněná varianta popsaného záznamu HTTPS
- SVCB obsahuje také název služby
 - ve formátu Attrleaf podle [RFC 8552](#) a [RFC 8553](#)
 - názvy služeb jsou přidělené v [registru v IANA](#)
- před název služby je možné vložit i nestadardní port

Příklad záznamu SVCB

```
_8443._foo.api.example.com. 7200 IN SVCB 0 svc4.example.net.
```

DNS záznam DDR

- Discovery of Designated Resolvers (DDR)
 - česky objevení **jmenovaných** resolverů
 - provozovatel původního resolveru jmenuje následovníka
- pro přechod na šifrovanou komunikaci s resolverem
 - určeno pro klienty
 - znají IP resolveru, chtějí přejít na šifrování
- provozovatel resolveru tak hlásí podporu šifrování
 - předpokládá se, že jde o jednoho provozovatele
- standard dle [RFC 9462](#)

Předání informací

- výchozí protokol DNS je nešifrovaný
- máme další možnosti
 - DNS-over-TLS (DoT) v [RFC 7858](#)
 - DNS-over-HTTPS (DoH) v [RFC 8484](#)
 - DNS-over-QUIC (DoQ) v [RFC 9250](#)
- musíme ovšem klientům nějak signalizovat
 - obvykle používáme DHCP a RA
 - tím ale předáme jen **IP adresu**
- moderní protokoly ale vyžadují víc
 - doménové jméno serveru (kvůli certifikátu)
 - adresu resolveru v rámci webového serveru (URI)
 - někdy chceme i nestandardní port a další

Dva mechanismy

- standard definuje dva mechanismy
 - oba využívají záznamu SVCB pro získání konfigurace
 - podle toho, zda známe IP adresu nebo doménové jméno
- ① když známe jen IP adresu resolveru
 - klient položí zvláštní typ dotazu (SUDN, dle [RFC 6761](#))
 - zeptá se na jméno `_dns.resolver.arpa.`
 - získá tak nešifrovanou cestou záznam SVCB s dalšími informacemi
- ② když známe doménové jméno resolveru
 - zeptáme se na záznam typu SCVB u jeho jména
 - dozvíme se tak o dalších možnostech komunikace

Potvrzení provázanosti

- klient potřebuje potvrdit spojitost mezi resolversy
 - aby nechtěně nezačal komunikovat s někým cizím
- oba mechanismy umožňují zkontrolovat provázání
 - buď oba resolversy běží na **stejně IP**
 - nebo existuje **certifikát** obsahující původní IP
- klient může kontrolu za určitých okolností vynechat
 - například při použití privátních či link-local IP
 - ty nemohou být samozřejmě v certifikátu
 - resolver by pak měl mít stejnou IP jako původní

Příklady záznamů

- záznamů může být samozřejmě více
- jejich pořadí je možné určit prioritou
- formát jmen určuje [samostatný dokument](#)

Příklad záznamu pro DoH, DoT a DoQ

```
_dns.example.net. 7200 IN SVCB 1 example.net. (  
  alpn=h2 dohpath=/dns-query{?dns} )
```

```
_dns.example.net. 7200 IN SVCB 1 dot.example.net (  
  alpn=dot port=8530 )
```

```
_dns.example.net. 7200 IN SVCB 1 doq.example.net (  
  alpn=doq port=8530 )
```

Když známe jen IP

- dotaz na SVCB jménem **_dns** s autoritou **resolver.arpa**
- odpovědi jsou podrobnosti týkající se šifrovaného resolveru
- v odpovědi by měly přijít i IP adresy resolveru
 - záznamy A a/nebo AAAA v Additional Answers
 - šifrovaný resolver může mít více IP než původní, ale ne méně
 - vlastně obdoba GLUE záznamu

Příklad záznamu pro DoH

```
_dns.resolver.arpa. 7200 IN SVCB 1 doh.example.net (  
  alpn=h2 dohpath=/dns-query{?dns} )
```


Kdo to podporuje?

- všechno jsou to velmi **mladé standardy**
- podpora v nástrojích je zatím **nekompletní**
 - na straně DNS resolverů problém není, lze nasadit
 - software (prohlížeče) umí obvykle jen část standardu
 - nebo umí starší variantu z některého návrhu
- na plnohodnotnou podporu si musíme počkat
 - ale už teď můžeme u svých domén použít

Příklad

```
$ dig https www.petrkrcomar.cz
www.petrkrcomar.cz. 3600 IN HTTPS 1 . alpn="h2" port=443
                        ipv4hint=37.205.10.41 ipv6hint=2a03:3b40:fe:50d::1
```

Kde to vyzkoušet HTTPS

- HTTPS si lze vyzkoušet na test.petrkrccmar.cz
 - dva porty 443 a 4433 s různým obsahem
 - obvykle přijdete na 443, záznam HTTPS vás ale pošle na 4433
 - Chrome podporuje jen částečně (jen alpn)
 - Firefox po zapnutí `network.dns.native_https_query`
 - Safari na macOS podporuje
- podporu Enhanced Client Hello (ECH) vyzkoušíte na tls-ech.dev
 - Chrome podporuje od verze 117
 - Firefox podporuje od verze 118
 - Safari na macOS nepodporuje

Kde vyzkoušet DDR

- podporu má zapnutou například [Cloudflare](#)
- Windows umí od vydání 22H2
- Apple umí v macOS Ventura a iOS 16
- Android umí od verze 11
- v Linuxu lze doplnit pomocí dnstool z PowerDNS

Příklad

```
$ dig svcb _dns.resolver.arpa @1.1.1.1
...
;; ANSWER SECTION:
_dns.resolver.arpa. 300 IN SVCB 1 one.one.one.one. alpn="h2,h3" port=443
    ipv4hint=1.1.1.1,1.0.0.1 ipv6hint=2606:4700:4700::1111,2606:4700:4700::1001
    key7="/dns-query{?dns}"
_dns.resolver.arpa. 300 IN SVCB 2 one.one.one.one. alpn="dot" port=853
    ipv4hint=1.1.1.1,1.0.0.1 ipv6hint=2606:4700:4700::1111,2606:4700:4700::1001
```

Otázky?

Petr Krčmář
petr.krcmar@iinfo.cz