zurück

Reverse Lookup – PTR-Records

Beim Reverse Lookup (auch **Reverse DNS** oder **rDNS**) wird eine IP-Adresse zur zugehörigen Domain aufgelöst. Während der normale DNS-Lookup **Domain → IP** lautet, erfolgt beim Reverse Lookup die Abfrage:

IP → Domain

Dies geschieht über **PTR-Records** (Pointer Records).

Warum Reverse Lookup?

Reverse Lookups werden verwendet für:

- Mailserver-Authentifizierung (SPAM-Schutz)
- Logging & Monitoring
- Netzwerkdiagnose (z. B. „Wer ist diese IP?“)
- Sicherheitsanalysen
- Troubleshooting

Beispiel:

```
IP 8.8.8.8 → dns.google.
```

Wie funktioniert Reverse DNS?

Statt einer Forward-Zone (z. B. `example.com`) gibt es Reverse-Zonen:

- IPv4 → unter `in-addr.arpa`
- IPv6 → unter `ip6.arpa`

Reverse Lookup für IPv4

IP-Adresse:

```
192.168.10.25
```

Die Reihenfolge wird **umgedreht**:

```
25.10.168.192.in-addr.arpa.
```

PTR-Record:

```
25.10.168.192.in-addr.arpa. IN PTR server1.example.com.
```

ASCII:

```
IP 192.168.10.25
→ Anfrage an 25.10.168.192.in-addr.arpa
→ Antwort: server1.example.com
```

Reverse Lookup für IPv6

IPv6 wird in Hex-Ziffern **umgedreht** und in `ip6.arpa` dargestellt.

Beispiel IPv6:

```
2001:db8::1
```

Umgedreht:

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
```

Nicht gerade schön, aber normiert.

Wer verwaltet Reverse DNS?

Wichtig:

- bei **öffentlichen IPs**: der Internet-Provider
    - bei **privaten Netzen**: du selbst / dein DNS-Server

  Das heißt:

- Für z. B. `192.168.0.x` richtest **du** die PTR-Records ein.
- Für `80.x.x.x` oder `88.x.x.x` (öffentliche IP): zuständig ist dein ISP.

Warum Mailserver ohne Reverse DNS scheitern

Mailserver prüfen:

1. Hat die IP einen PTR-Eintrag?
2. Zeigt der PTR auf eine gültige Domain?
3. Passt Forward Lookup wieder zurück?

Beispiel eines gültigen Setups:

```
203.0.113.5 → mail.example.com  (PTR)
mail.example.com → 203.0.113.5  (A)
```

Fehlt einer der Schritte → viele Mailserver lehnen E-Mails ab.

Beispiele Reverse Lookup per dig

IPv4

```
dig -x 8.8.8.8
```

Ausgabe:

```
8.8.8.8.in-addr.arpa. PTR dns.google.
```

IPv6

```
dig -x 2001:4860:4860::8888
```

Reverse DNS Zonenstruktur

Beispiel für 192.168.10.0/24:

```
$ORIGIN 10.168.192.in-addr.arpa.

25  IN PTR  host1.example.com.
50  IN PTR  printer.example.com.
70  IN PTR  nas.example.com.
```

PTR-Record Aufbau

```
IP-last-octet    IN PTR   hostname.
```

Beispiel:

```
24 IN PTR client24.lan.example.com.
```

Sicherheitshinweise

- PTR verrät oft interne Hostnamen → vorsichtig in öffentlichen Netzen
    - interne Reverse DNS sollte logisch und klar benannt sein
      * Reverse Einträge helfen bei Incident Response („wer war 192.168.10.70?“)

Kurzer Vergleich Forward vs Reverse

Vorgang	Typ	Beispiel
———————-	———–	———————————–
Domain → IP	Forward	example.com → 93.184.216.34
IP → Domain	Reverse	8.8.8.8 → dns.google.
Record-Typ	Forward	A / AAAA
Record-Typ	Reverse	PTR

Zusammenfassung

- Reverse Lookup = IP → Domain
    - IPv4 nutzt `in-addr.arpa`, IPv6 `ip6.arpa`
      * PTR-Records weisen IP-Adressen einem Hostnamen zu
      * wichtig für Mailserver, Logs, Security, Forensik
      * öffentliche PTRs verwaltet dein Provider