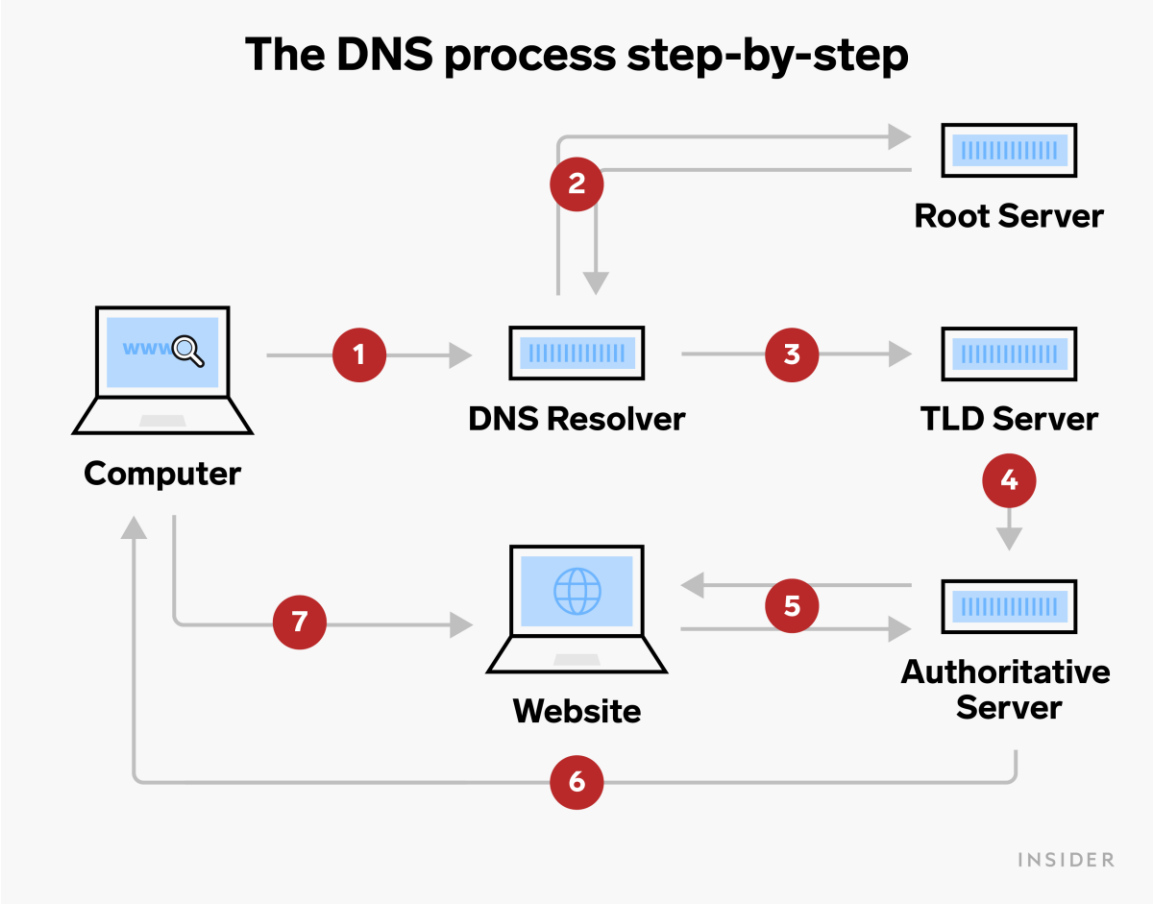
DNS Server Study Guide

Mahmoud Omar Abdelghaffar

https://www.linkedin.com/in/mahmoud-omar-022153214

1. DNS Servers

DNS servers are categorized into several types to manage and resolve domain names across the Internet.



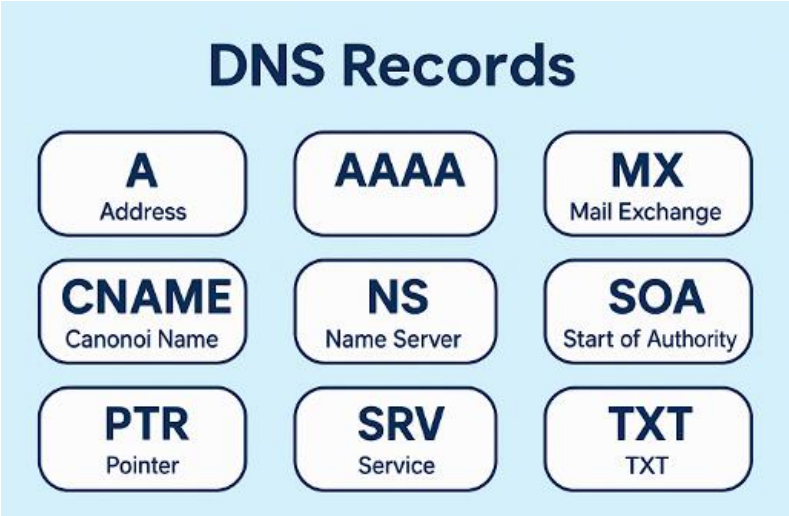
- **Root Server:** The top of the DNS hierarchy, knows locations of all TLD servers.

- **TLD Server:** Top-Level Domain servers manage domains like .com, .net, etc.

- **Authoritative Name Server**: Holds actual DNS records and responds to queries with answers.

2. DNS Records

DNS records are used to map domain names to IP addresses and other resources.



- A – Maps a domain to an IPv4 address.

- AAAA – Maps a domain to an IPv6 address.

- CNAME – Canonical name record, alias of one domain to another.

- MX – Mail exchange record; routes emails to correct server with priority.

- NS – Name server record indicating authoritative servers.

- SOA – Start of Authority, administrative information (primary NS, contact email, serial number).

- SRV – Service locator; defines location (port, hostname) of servers for specific services.

- PTR – Pointer record, for reverse DNS lookups.

- TXT – Holds arbitrary text or structured data like SPF records.

3. Roles of Name Servers

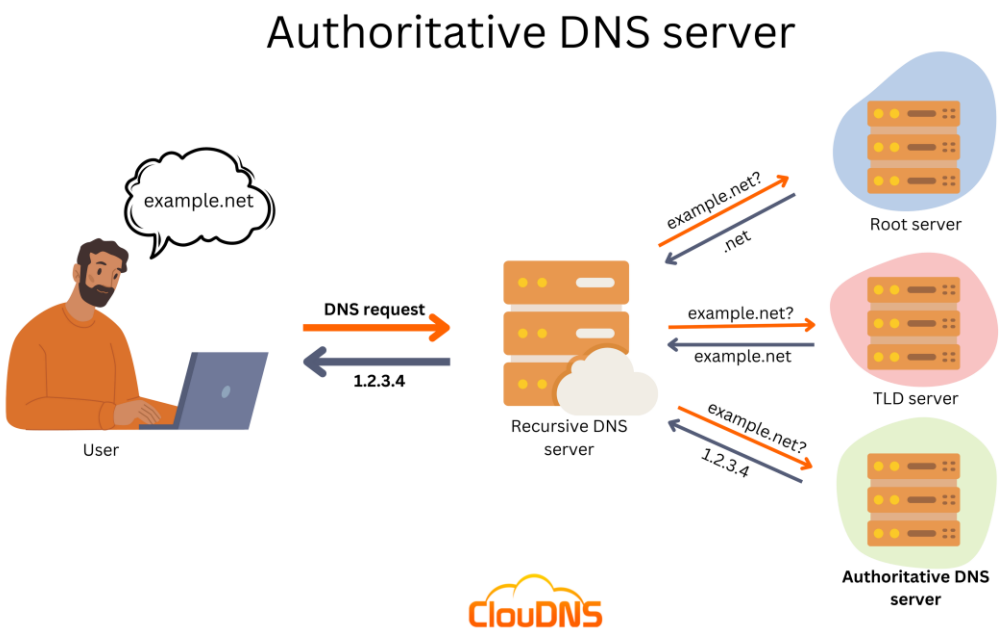
- **Caching-Only**: Stores responses temporarily; does not hold original DNS data.

```
tecmint@dns:~
[tecmint@dns ~]$
[tecmint@dns ~]$ dig facebook.com

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6_4.6 <<>> facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61554
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;facebook.com.                  IN      A

;; ANSWER SECTION:
facebook.com.           900     IN      A       173.252.110.27

;; AUTHORITY SECTION:
facebook.com.           172799  IN      NS      a.ns.facebook.com.
facebook.com.           172799  IN      NS      b.ns.facebook.com.

;; ADDITIONAL SECTION:
a.ns.facebook.com.      172799  IN      A       69.171.239.12
b.ns.facebook.com.      172799  IN      A       69.171.255.12

;; Query time: 1294 msec
;; SERVER: 192.168.0.200#53(192.168.0.200)
;; WHEN: Thu Sep  4 04:33:02 2014
;; MSG SIZE  rcvd: 113
```

```
[tecmint@dns ~]$ dig facebook.com

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6_4.6 <<>> facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15311
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;facebook.com.                  IN      A

;; ANSWER SECTION:
facebook.com.           898     IN      A       173.252.110.27

;; AUTHORITY SECTION:
facebook.com.           172797  IN      NS      a.ns.facebook.com.
facebook.com.           172797  IN      NS      b.ns.facebook.com.

;; ADDITIONAL SECTION:
a.ns.facebook.com.      172797  IN      A       69.171.239.12
b.ns.facebook.com.      172797  IN      A       69.171.255.12

;; Query time: 0 msec
;; SERVER: 192.168.0.200#53(192.168.0.200)
;; WHEN: Thu Sep  4 04:33:04 2014
;; MSG SIZE  rcvd: 113

[tecmint@dns ~]$
```

```
tecmint@dns:~
[tecmint@dns ~]$
[tecmint@dns ~]$ nslookup facebook.com
Server:         192.168.0.200
Address:        192.168.0.200#53

Non-authoritative answer:
Name:   facebook.com
Address: 173.252.110.27

[tecmint@dns ~]$
```

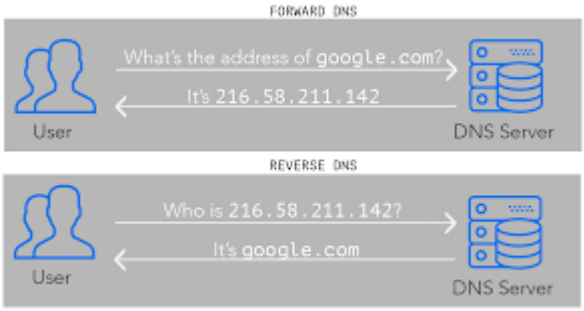
- **Authoritative:** Stores DNS records locally and answers with original data.



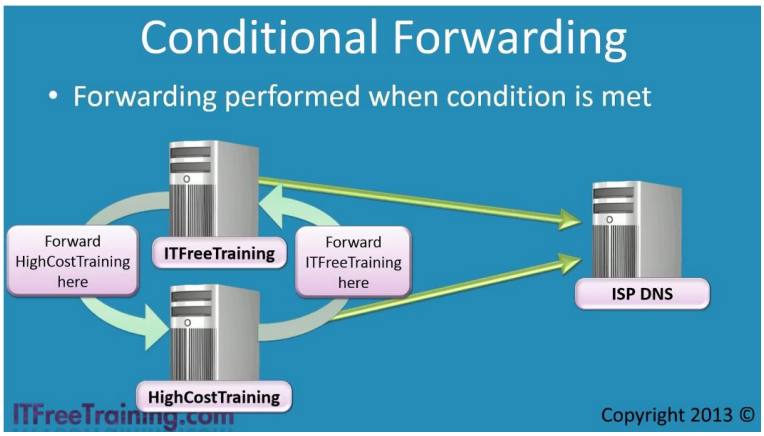
4. Types of DNS Zones

- **Forward Lookup Zone**: Resolves domain names to IP addresses.

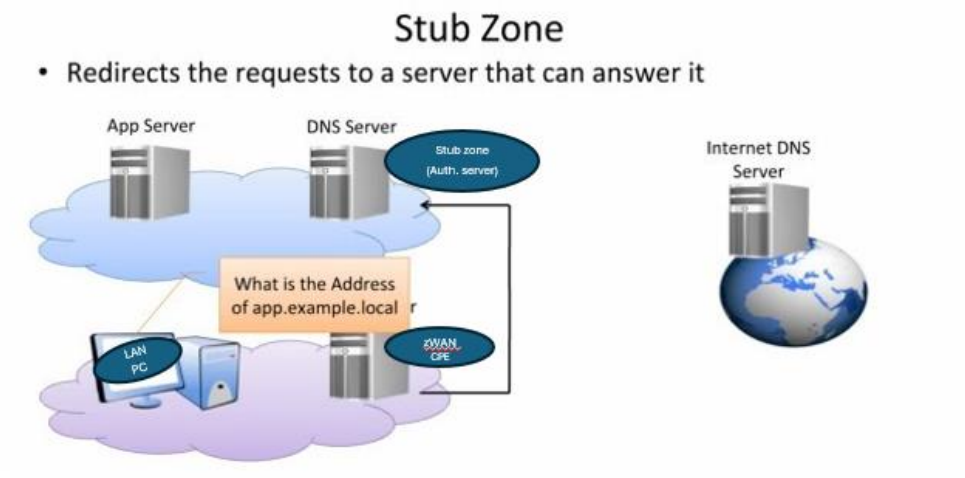
- **Reverse Lookup Zone**: Resolves IP addresses to domain names.



- **Conditional Forwarder Zone**: Forwards queries to specific DNS servers based on domain. It simply passes the queries to the designated name servers and returns



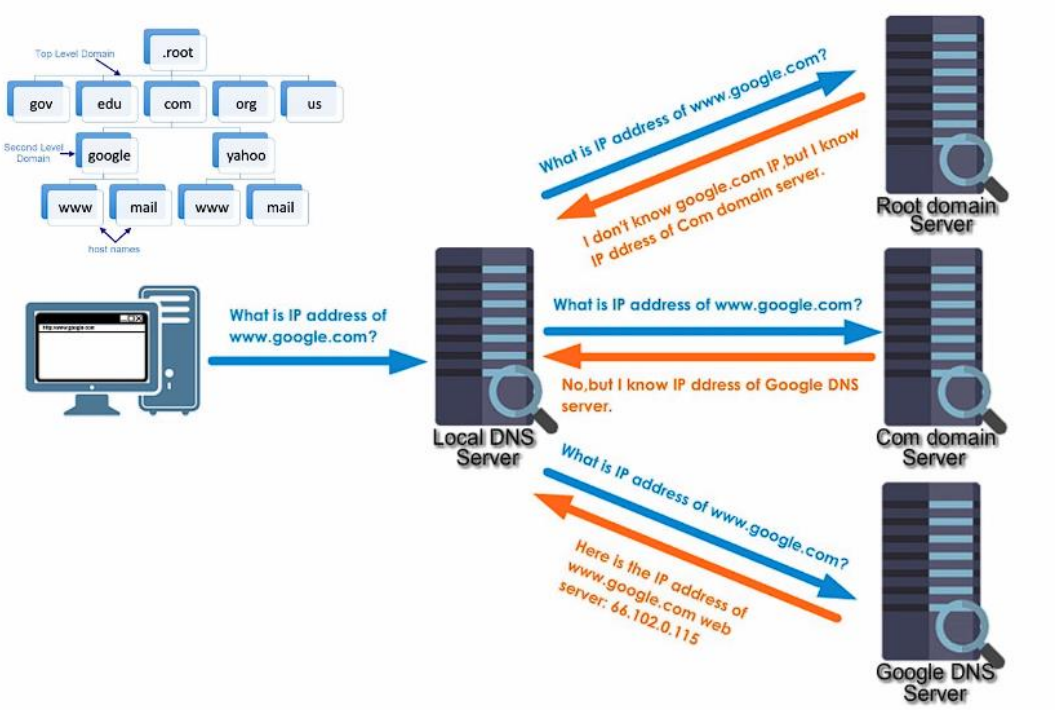
- Stube Zone: maintains only the necessary information to contact the authoritative DNS servers, including NS records and A records required for resolution



5. DNS Query Types

1. **Recursive queries**
    - Demand a name resolution or the answer "it can't be found"
    - Is between DNS client and its local DNS Server

2. **Iterative queries**
    - Do not demand a name resolution, which means other DNS servers may provide a name resolution if they know or simply respond with a referral
    - It between a local DNS server and other DNS servers



3. Non-Recursive queries
    - is expected to provide the answer directly, either from its local cache or from its authoritative zone data. If the server does not have the requested information, it will respond with an empty or negative response,
    - indicating that it cannot resolve the query, rather than forwarding the request to other DNS servers

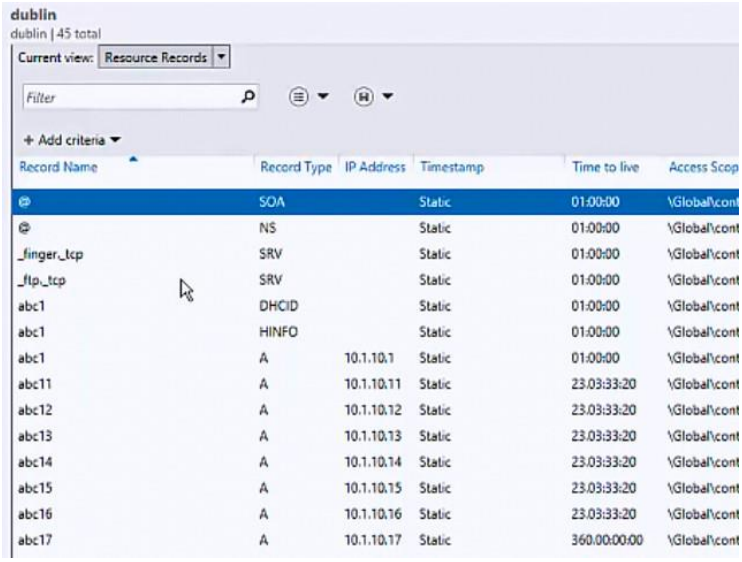
6. Resource Record (RR) Syntax

RR Syntax Format:

<domain-name> <TTL> <class> <type> <RDATA>

Example: example.com. 3600 IN A 192.0.2.1

```
[{'name':'junebugweddings.com.,',
  'ttl': 300,
  'type': 1,
  'class': 1,
  'data': '104.27.191.148'},
 {'name': 'junebugweddings.com.,',
  'ttl': 300,
  'type': 1,
  'class': 1,
  'data': '104.27.190.148'}]
```



7. Zone Transfer

There are two modes of zone transfer over TCP are implemented

1. Full Zone Transfer (AXFR)

- **AXFR** stands for **Asynchronous Full Zone Transfer**.
- It transfers **the entire DNS zone file** from the master to the slave server.
- Used when:
    - The slave server is syncing for the first time.
    - There are **major changes** or **no incremental data** available.

Key Characteristics:

- Transfers all records (A, MX, CNAME, etc.).
- Slower and uses more bandwidth than IXFR.
- Triggered manually or when the serial number changes.

2. Incremental Zone Transfer (IXFR)

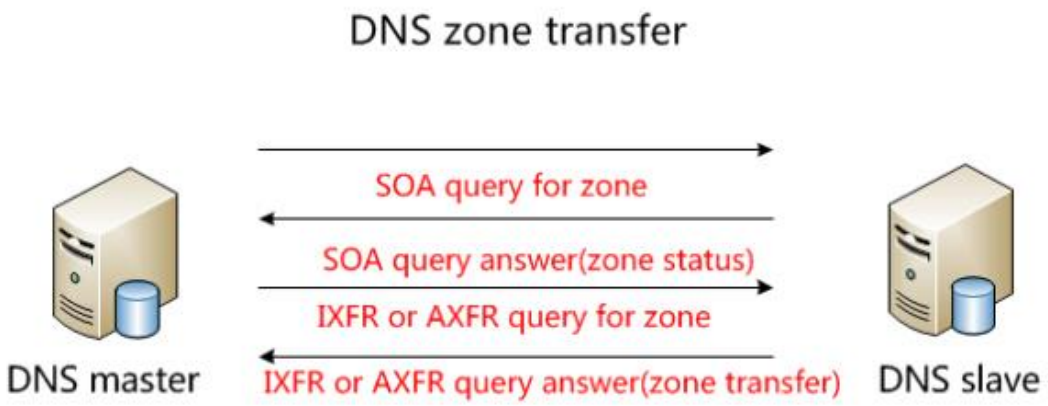
- **IXFR** stands for **Incremental Zone Transfer**.
- Only transfers the **changed DNS records** (deltas) since the last update.
- Requires both servers to support IXFR and have a shared change history.

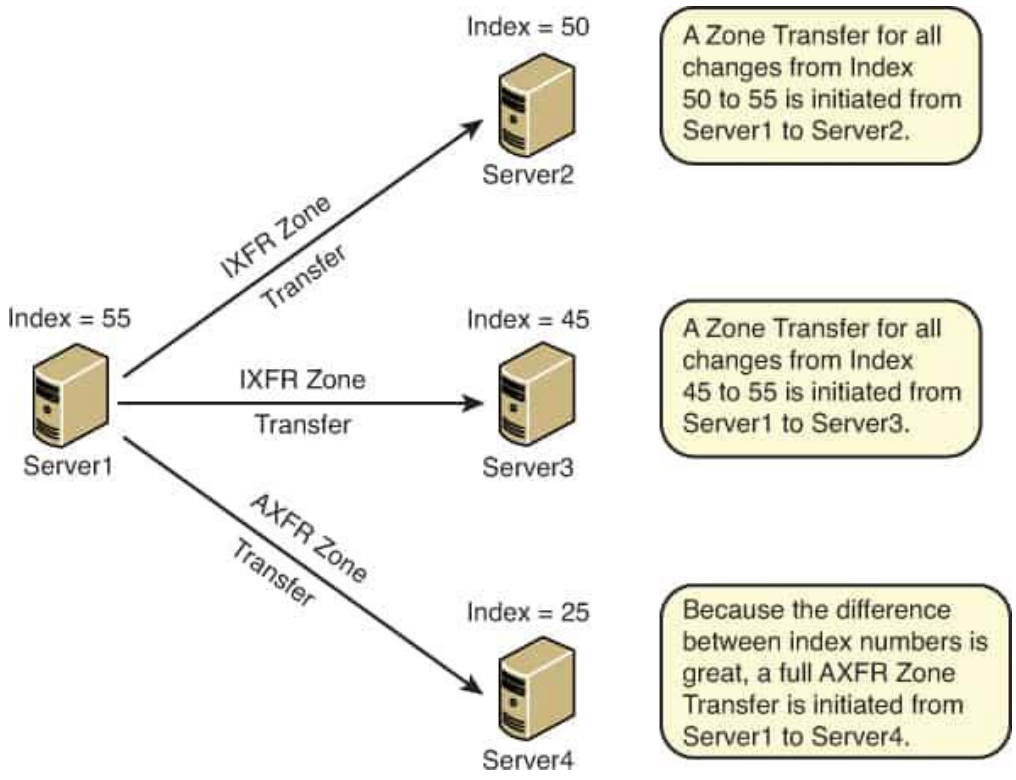
Key Characteristics:

- More efficient than AXFR.
- Faster, uses less bandwidth.
- Ideal for frequently updated zones.

Summary Table

Feature	**AXFR (Full)**	**IXFR (Incremental)**
Transfer Type	Entire zone	Only changed records
Efficiency	Lower	Higher
Use Case	Initial sync, major updates	Frequent minor changes
Supported by	Most DNS servers	Only newer DNS servers





8. DNS Deployment Model

1. BIND (Berkeley Internet Name Domain)

- **Most widely used** DNS server on the internet.
- Developed by: Internet Systems Consortium (ISC)
- Supports:
    - **Authoritative DNS**
    - **Recursive DNS**
    - **Caching**
    - **DNSSEC**
    - **Zone transfers (AXFR/IXFR)**

Use Case:
Full-featured DNS server for both authoritative and recursive needs.

Pros:

- Very flexible and customizable.

- Large community and documentation.
- Supports dynamic updates and views.

Cons:

- More complex to configure.
- Larger attack surface due to many features.

---

2. NSD (Name Server Daemon)

- **Authoritative-only DNS server** (no recursion).
- Developed by: NLnet Labs
- Lightweight, high performance.

Use Case:
High-performance authoritative DNS server (e.g., for root/TLD servers).

Pros:

- Simple and secure.
- Very fast for authoritative responses.
- Lower resource usage.

Cons:

- Does **not** support recursion or caching.
- Must be paired with a separate resolver (e.g., Unbound).

---

3. Unbound

- **Recursive DNS resolver only** (no authoritative zones).
- Developed by: NLnet Labs
- Fast, secure, modern DNS resolver.

Use Case:
Caching recursive DNS server for internal or local networks.

Pros:

- High security (DNSSEC validation by default).
- Lightweight and easy to configure.
- Supports DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH).

Cons:

- Cannot serve authoritative DNS zones.

Typical DNS Deployment Model

Many modern setups separate roles:

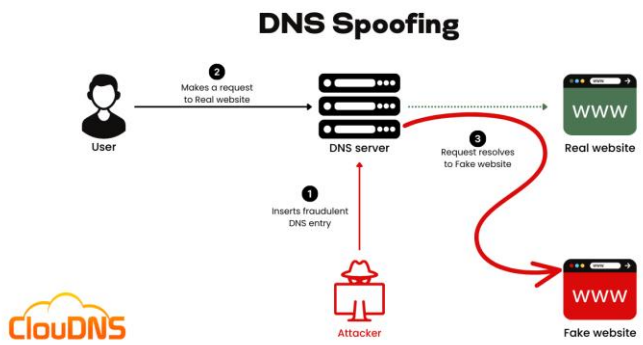
- **NSD** for authoritative responses
- **Unbound** for internal recursive resolution
- Or use **BIND** for both in small/medium systems

Engine	Main Config File	Zone Files Used?	Zone File Location	Control Keys
BIND	/etc/bind/named.conf	✅ Yes	/var/cache/bind/, /etc/bind/zones	rndc.key
NSD	/etc/nsd/nsd.conf	✅ Yes	/etc/nsd/zones/	nsd_control.key
Unbound	/etc/unbound/unbound.conf	❌ No (uses local-data)	N/A (uses directives)	unbound_control.key

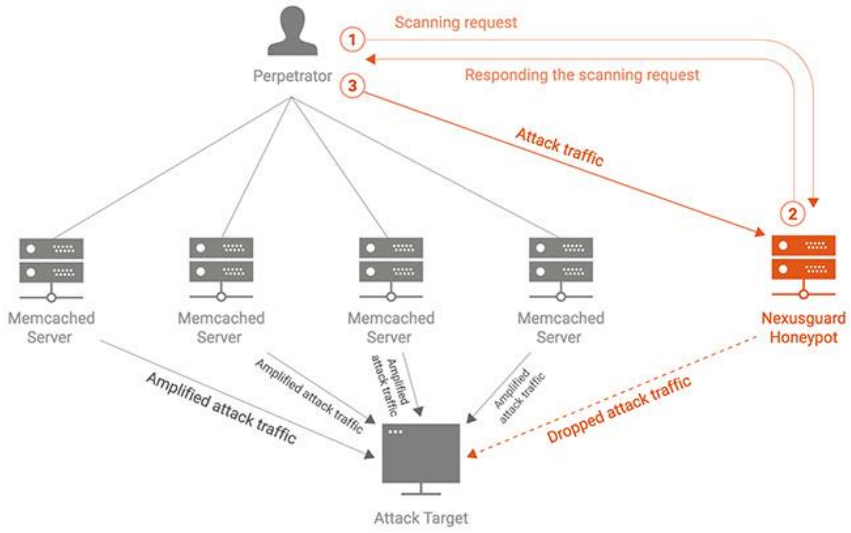
9. DNS Attacks

DNS can be a target for various types of attacks:

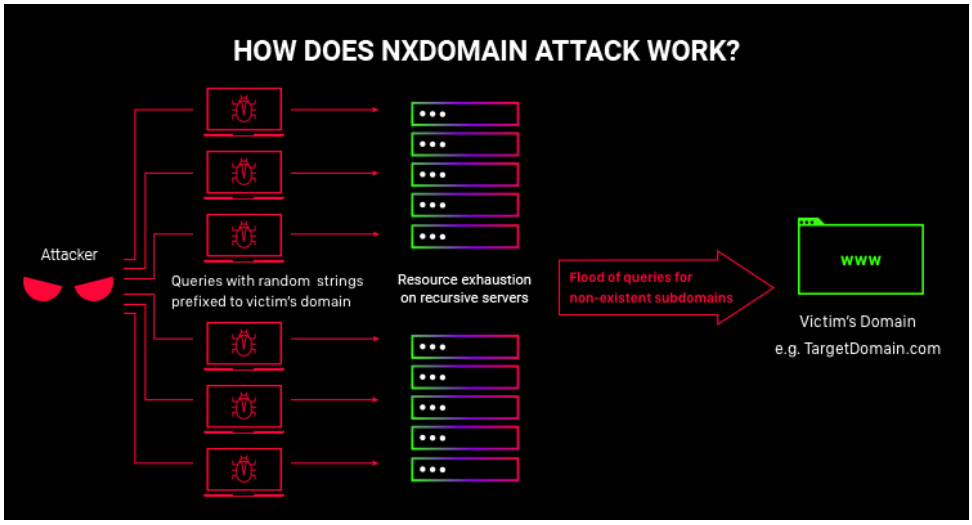
- **DNS Spoofing/Poisoning** – Redirecting traffic to malicious sites.



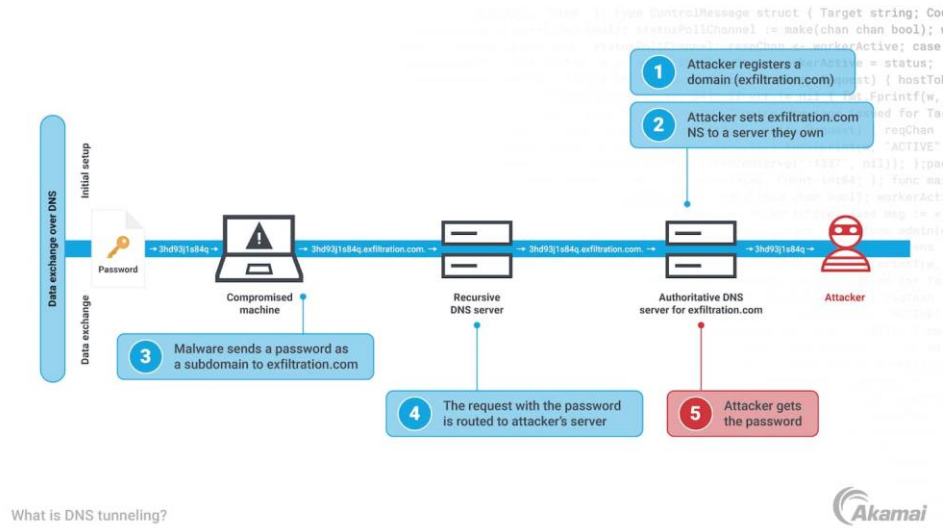
- **DNS Amplification** – DDoS using spoofed queries to produce large responses.



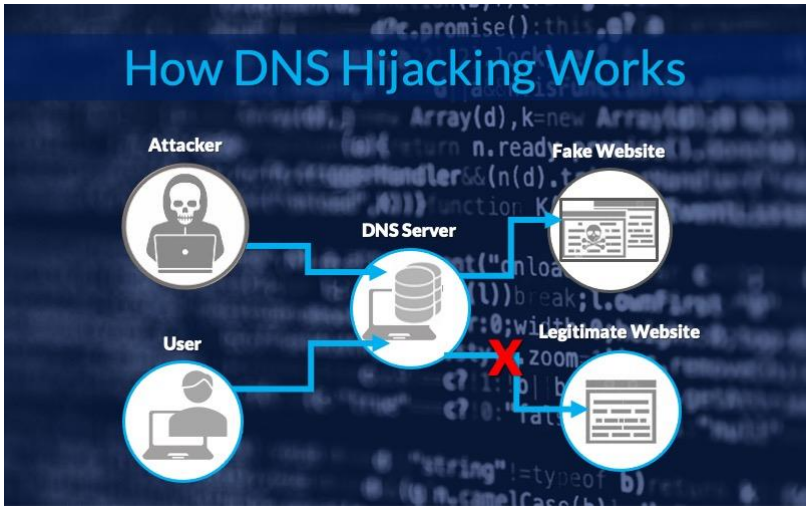
- **NXDOMAIN Attack** – Floods DNS with requests for nonexistent domains.



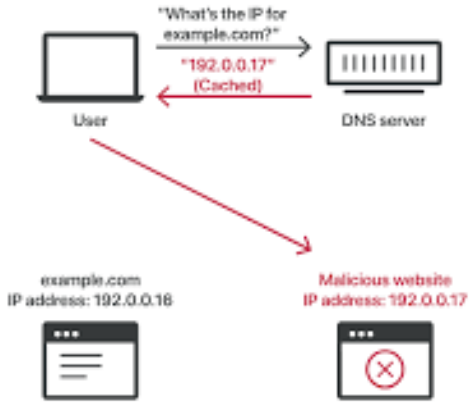
- **DNS Tunneling** – Encodes data in DNS queries/responses to bypass firewalls.



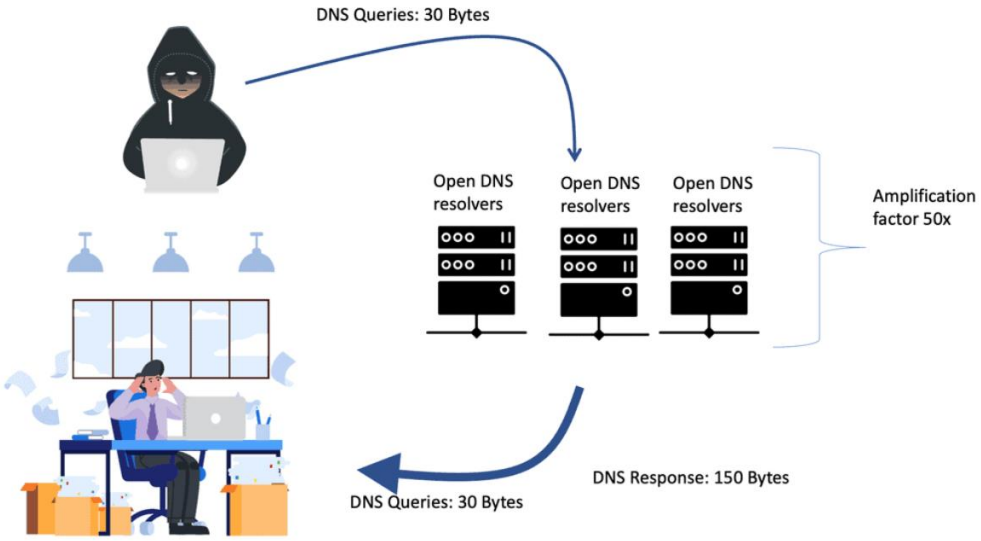
- **Domain Hijacking** – Takes control over a registered domain.



- **Cache Poisoning** – Inserts false data into DNS cache.



- **DNS Reflection** – Attacker spoofs victim's IP to trigger large DNS replies.



10. DNS Security Best Practices

- Use DNSSEC to authenticate responses.

- Restrict zone transfers to authorized IPs.

- Regularly patch DNS software.

- Configure rate-limiting and logging.

- Use split-horizon DNS where applicable.

- Avoid open resolvers unless necessary.

- Monitor DNS traffic for anomalies.