Flashcards — What Happens When You Hit Enter

Episode 002 · cut along the lines, or import NetSecViz-002-Flashcards.csv into Anki (Front/Back).

Q What does DNS do?
A Maps a human domain name to a numeric IP address.

Q Order of DNS lookup locations?
A Browser cache -> OS cache -> recursive resolver -> root -> TLD -> authoritative.

Q Recursive resolver vs authoritative server?
A Recursive does the full lookup for you; authoritative holds the real record for the domain.

Q Iterative vs recursive in DNS?
A The resolver's walk down the hierarchy is iterative (referrals); the service it gives you is recursive.

Q Three tiers of the DNS hierarchy?
A Root -> TLD -> authoritative.

Q What does a root server do?
A It does not hold the answer; it points to the right TLD server.

Q What does a TLD server do?
A It points to the authoritative server for the name (e.g. the .net server).

Q What is the authoritative server?
A The one server that holds a domain's real DNS records.

Q What is 8.8.8.8?
A Google's public recursive resolver.

Q What is a TTL?
A Time To Live: how long a cached DNS answer stays valid.

Q Why does caching matter for security?
A A poisoned answer persists in the cache for the whole TTL, affecting every user of that resolver.

Q A vs AAAA record?
A A record holds an IPv4 address; AAAA holds IPv6.

Q What does a CNAME record do?
A Points one domain name to another name.

Q What transport does classic DNS use, and why does it matter?
A UDP: fast and connectionless with no handshake, so a reply is trusted with almost no verification.

Q How does classic DNS match a reply to a query?
A By a single 16-bit transaction ID.

Q How many possible transaction IDs are there?
A 65,536 (16 bits).

Q What does the resolver trust?
A The first reply that comes back with the matching transaction ID; it does not check who sent it.

Q What is DNS cache poisoning?
A Injecting a forged answer into a resolver's cache so it returns an attacker-controlled IP.

Q Explain cache poisoning as a 'race'.
A The attacker floods forged replies, each guessing the transaction ID, trying to beat the real server's reply.

Q What did Dan Kaminsky add in 2008?
A Looking up random made-up subdomains, giving unlimited fresh races, which made poisoning fast and reliable.

Q CVE number for the Kaminsky bug?
A CVE-2008-1447.

Q Does the attacker hack the DNS server?
A No. They trick the resolver into caching a forged answer. The protocol's trust is the flaw.

Q What happens to users after a successful poisoning?
A Everyone using that resolver is silently sent to the attacker's server until the TTL expires.

Q Does poisoning steal a password by itself?
A No; it redirects you to the wrong address, trusted completely. That sets up phishing or malware.

Q First defense against poisoning?
A Source-port randomization: an attacker must guess the transaction ID AND a random source port.

Q What does port randomization do to the guess space?
A Turns roughly 65,000 guesses into billions, killing the easy attack.

Q What is the complete fix for poisoning?
A DNSSEC: cryptographically signed records, so a forged answer fails the signature and is rejected.

Q Why isn't DNSSEC enough in practice?
A Adoption is low: not every domain signs, and not every resolver validates signatures.

Q Does DNSSEC encrypt DNS?
A No. DNSSEC provides integrity/authenticity (rejects forgeries); it does not hide the query.

Q DNSSEC vs DNS over HTTPS (DoH)?
A DNSSEC signs records to stop forgery; DoH encrypts the query in transit for privacy. Different goals.

Q Common DNS misconception?
A That DNS is queried fresh every time. Caching means most lookups never leave your resolver.

Q DNS's original sin, in one line?
A It trusts the first answer that looks right, which is exactly what cache poisoning abuses.