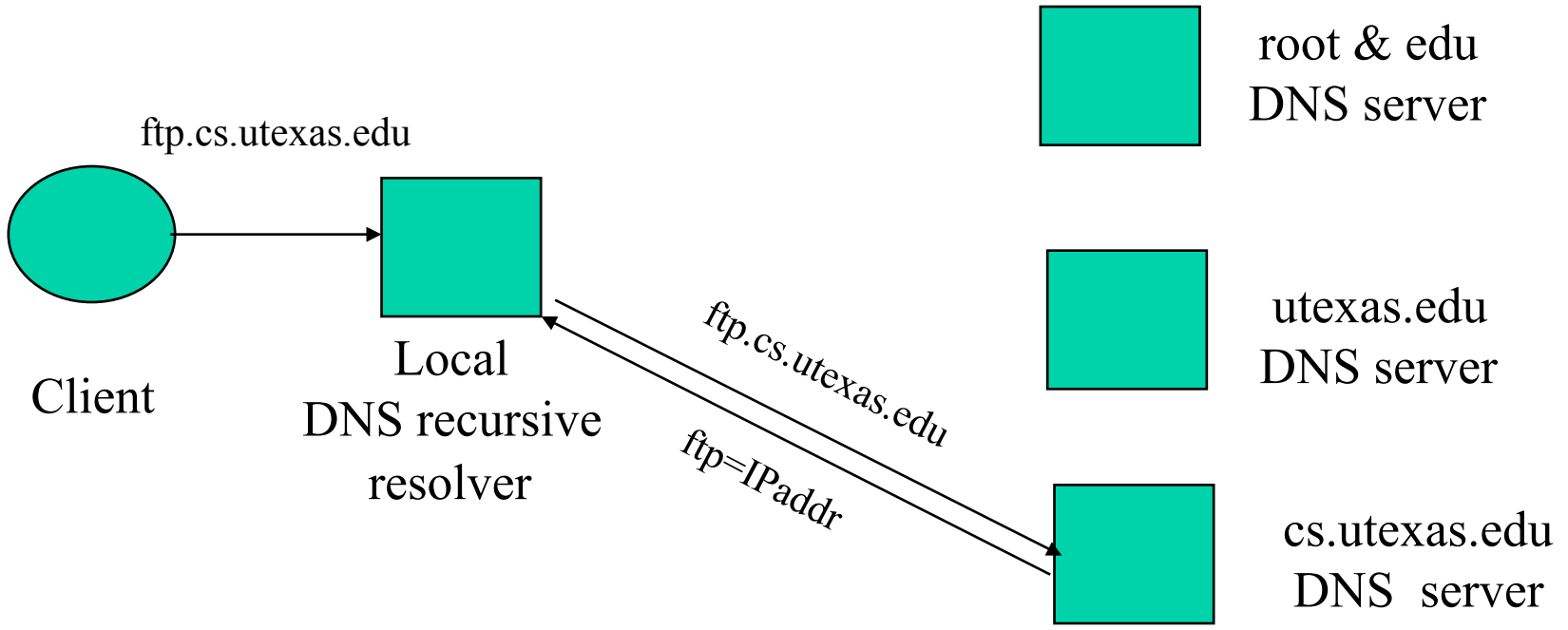
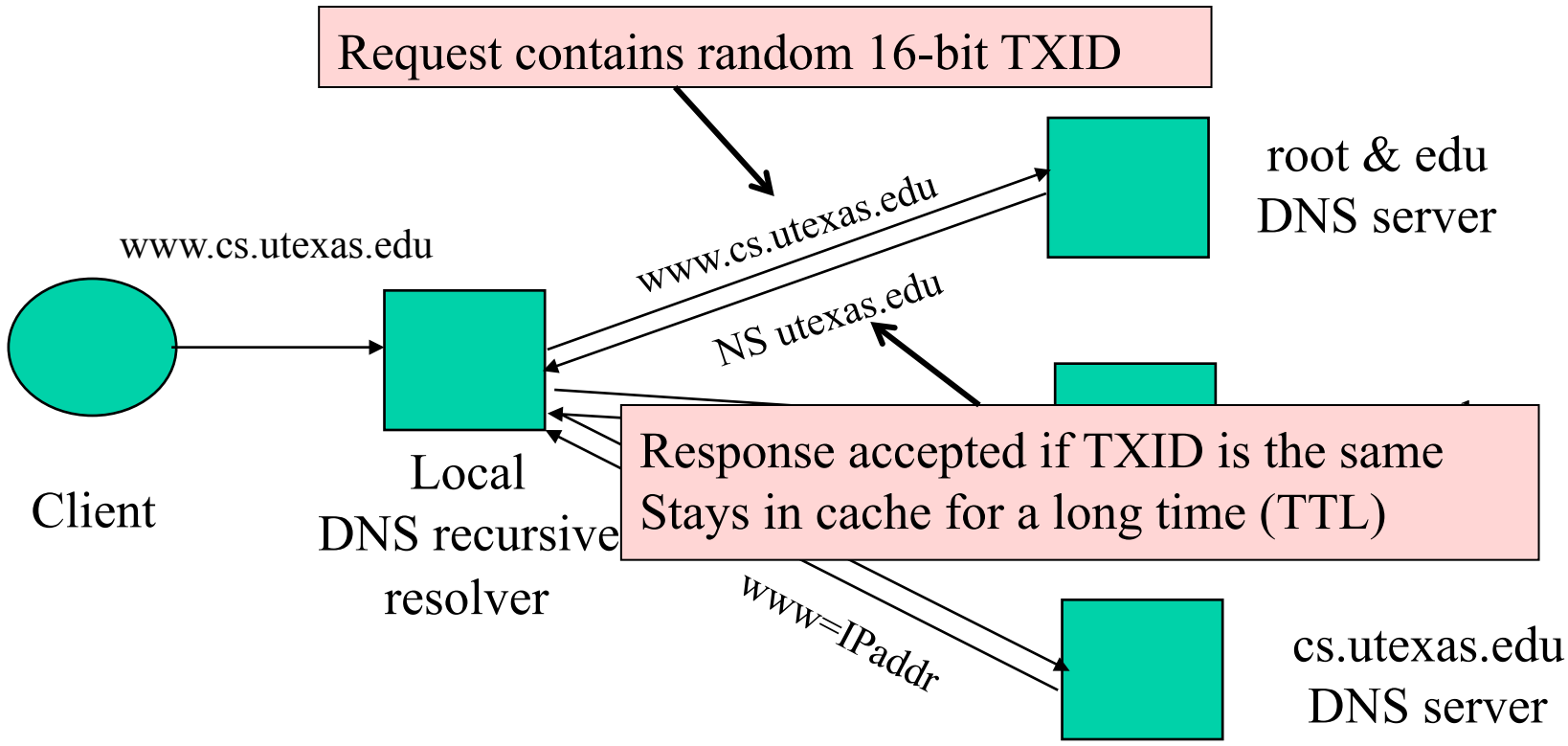
DNS Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Other queries may reuse some parts of lookup
    - NS records for domains
- DNS negative queries are cached
  - Don't have to repeat past mistakes
    - For example, misspellings
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record

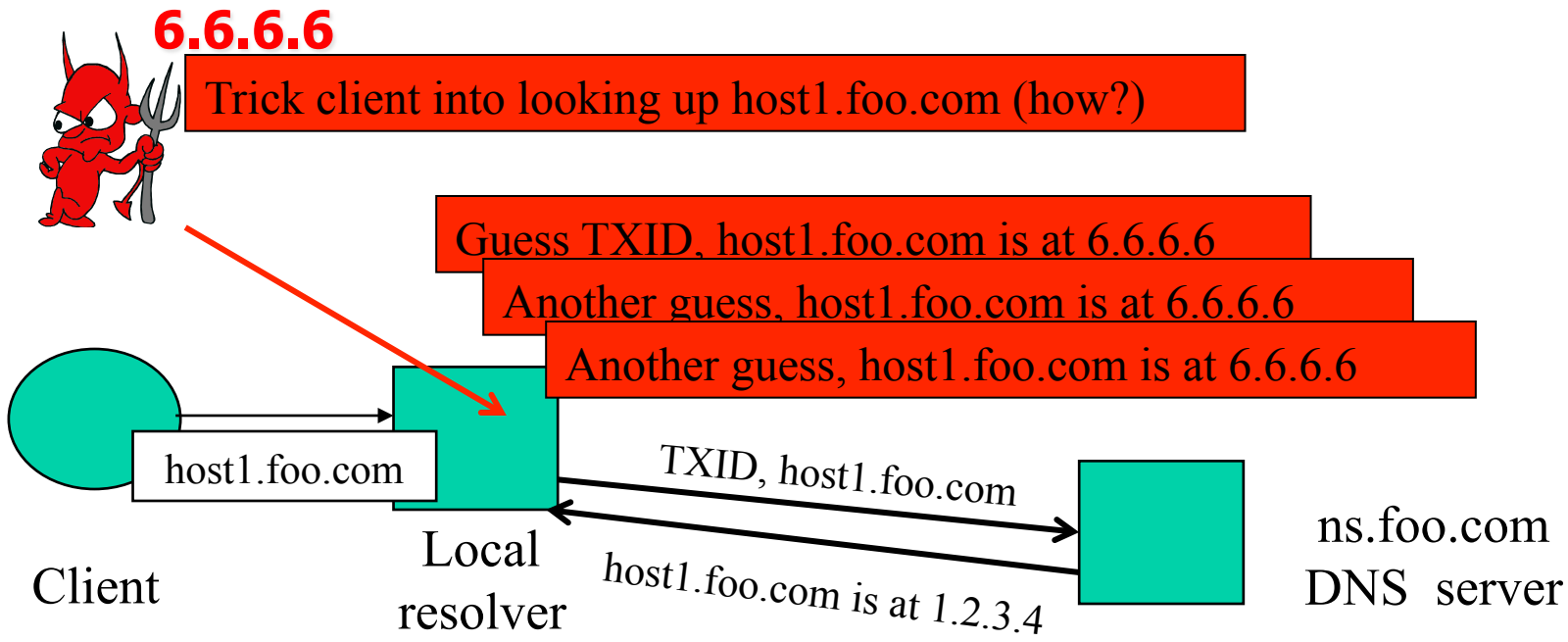
Cached Lookup Example



DNS "Authentication"



DNS Spoofing



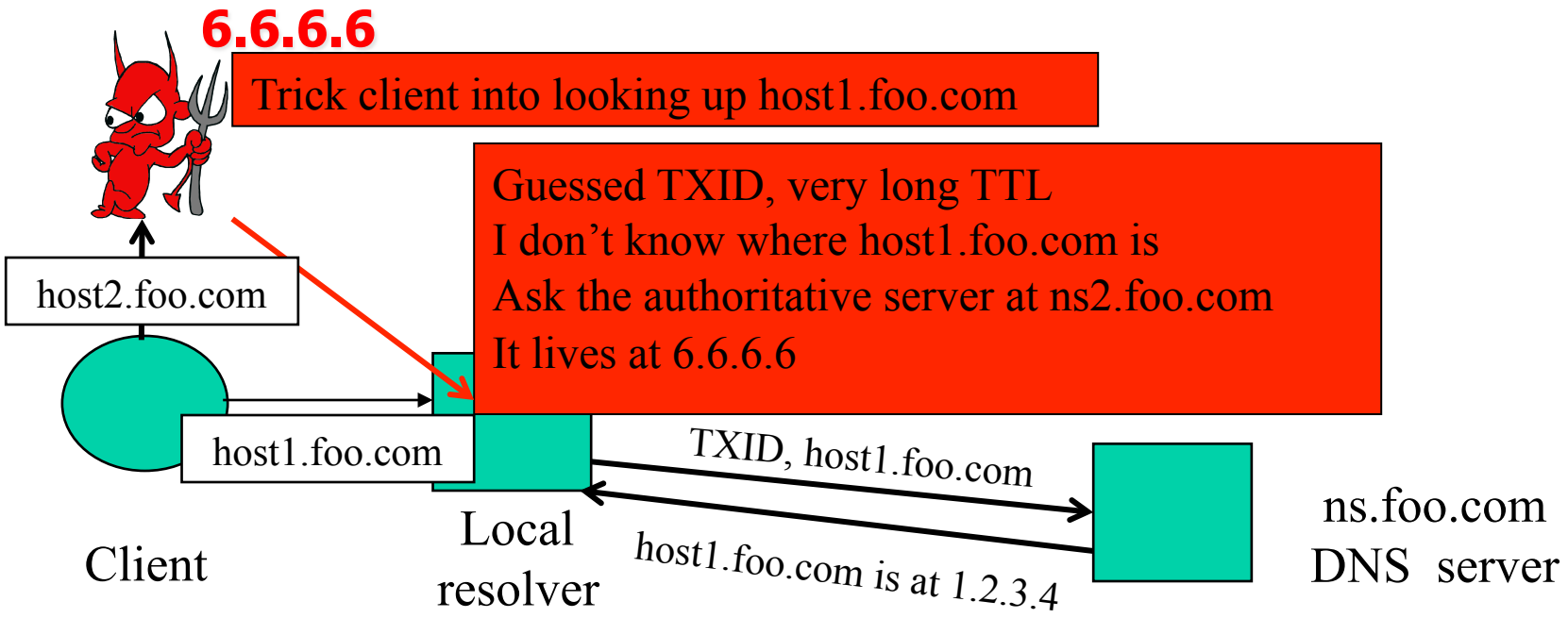
Several opportunities to win the race

If attacker loses, has to wait until TTL expires

… but can try again with host2.foo.com, host3.foo.com, etc.

… but what's the point of hijacking host3.foo.com?

Exploiting Recursive Resolving



If attacker wins, all future DNS requests will go to 6.6.6.6

The cache is now poisoned… for a very long time!

No need to win future races!

Triggering DNS Lookup

- Any link, any image, any ad, anything can cause a DNS lookup
  - No Javascript required, though it helps
- Mail servers will look up what bad guy wants
  - On first greeting: HELO
  - On first learning who they're talking to: MAIL FROM
  - On spam check (oops!)
  - When trying to deliver a bounce
  - When trying to deliver a newsletter
  - When trying to deliver an actual response from an actual employee

Reverse DNS Spoofing

- Trusted access is often based on host names
  - E.g., permit all hosts in .rhosts to run remote shell
- Network requests such as rsh or rlogin arrive from numeric source addresses
  - System performs reverse DNS lookup to determine requester's host name and checks if it's in .rhosts
- If attacker can spoof the answer to reverse DNS query, he can fool target machine into thinking that request comes from an authorized host
  - No authentication for DNS responses and typically no double-checking (numeric → symbolic → numeric)

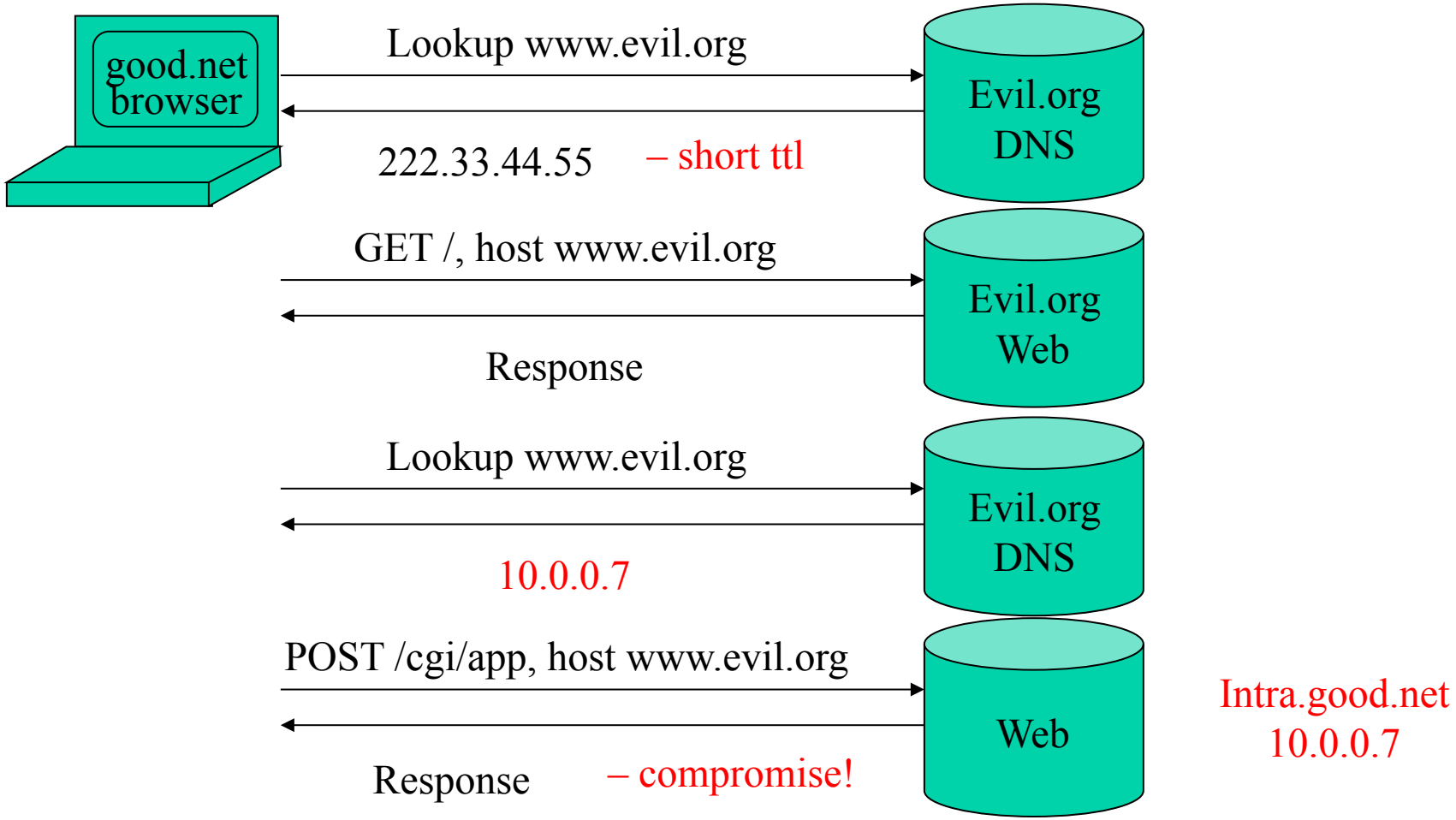
Pharming

- Many anti-phishing defenses rely on DNS
- Can bypass them by poisoning DNS cache and/or forging DNS responses
  - Browser: give me the address of www.paypal.com
  - Attacker: sure, it's 6.6.6.6 (attacker-controlled site)
- Dynamic pharming
  - Provide bogus DNS mapping for a trusted server, trick user into downloading a malicious script
  - Force user to download content from the real server, temporarily provide correct DNS mapping
  - Malicious script and content have the same origin!

JavaScript/DNS Intranet attack (I)

- Consider a Web server intra.good.net
  - IP: 10.0.0.7, inaccessible outside good.net network
  - Hosts sensitive Web applications
- Attacker at evil.org gets good.net user to browse www.evil.org
- Places Javascript on www.evil.org that accesses sensitive application on intra.good.net
  - This doesn't work because Javascript is subject to the "same-origin" policy
  - … but the attacker controls evil.org DNS

JavaScript/DNS Intranet attack (II)



Other DNS Vulnerabilities

- DNS implementations have vulnerabilities
  - Reverse query buffer overrun in old releases of BIND
  - MS DNS for NT 4.0 crashes on chargen stream
- Denial of service
  - Oct '02: ICMP flood took out 9 root servers for 1 hour
- Can use "zone transfer" requests to download DNS database and map out the network
  - "The Art of Intrusion": NYTimes.com and Excite@Home
  - Solution: block port 53 on corporate name servers

See http://cr.yp.to/djbdns/notes.html

Solving the DNS Spoofing Problem

- Long TTL for legitimate responses
  - Does it really help?
- Randomize port in addition to TXID
  - 32 bits of randomness, makes it harder for attacker to guess TXID
- DNSSEC
  - Cryptographic authentication of host-address mappings

DNSSEC

- Goals: authentication and integrity of DNS requests and responses
- PK-DNSSEC (public key)
  - DNS server signs its data (can be done in advance)
  - How do other servers learn the public key?
- SK-DNSSEC (symmetric key)
  - Encryption and MAC: $E_k(m, MAC(m))$
  - Each message contains a nonce to avoid replay
  - Each DNS node shares a symmetric key with its parent
  - Zone root server has a public key (hybrid approach)