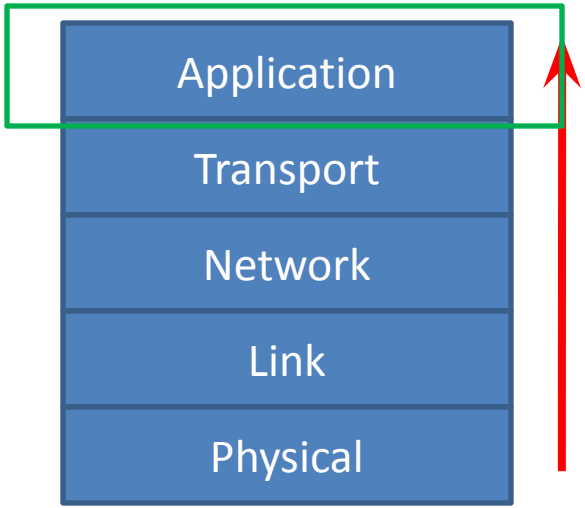
Computer and Network Security: Application Layer Attacks and Solutions

Kameswari Chebrolu

Outline

- Attacks at different layers of the protocol stack
- Solutions to the same



Only Protocol Specific Attacks relevant here

Application Layer Role

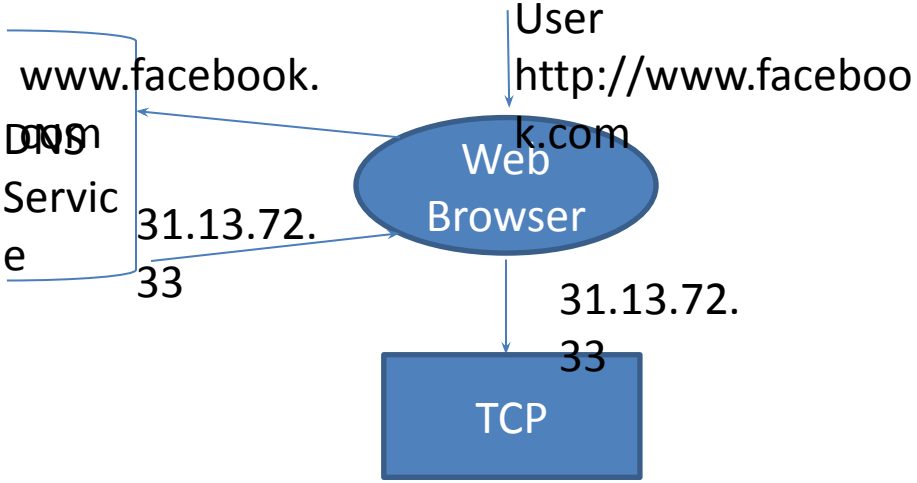
- Network infrastructure in place to enable variety of applications
  - Can transfer packets from a process on a given host to another process on another host
- Role of application developers:
  - Develop interesting/useful applications
  - Understand the building blocks and their interaction
  - Make the right choices and implement required functionality

Application Protocols

Application	Protocol	Transport
E-mail (covered later)	**SMTP (RFC 2821)**	**TCP**
Remote terminal access	Telnet (RFC 854)	TCP
Web (covered later)	**HTTP (RFC 2616)**	**TCP**
File Transfer	FTP (RFC 959)	TCP
Streaming Multimedia	Proprietary	TCP or UDP
Internet Telephony	Proprietary	Often UDP
Domain Name System	**DNS**	**UDP**

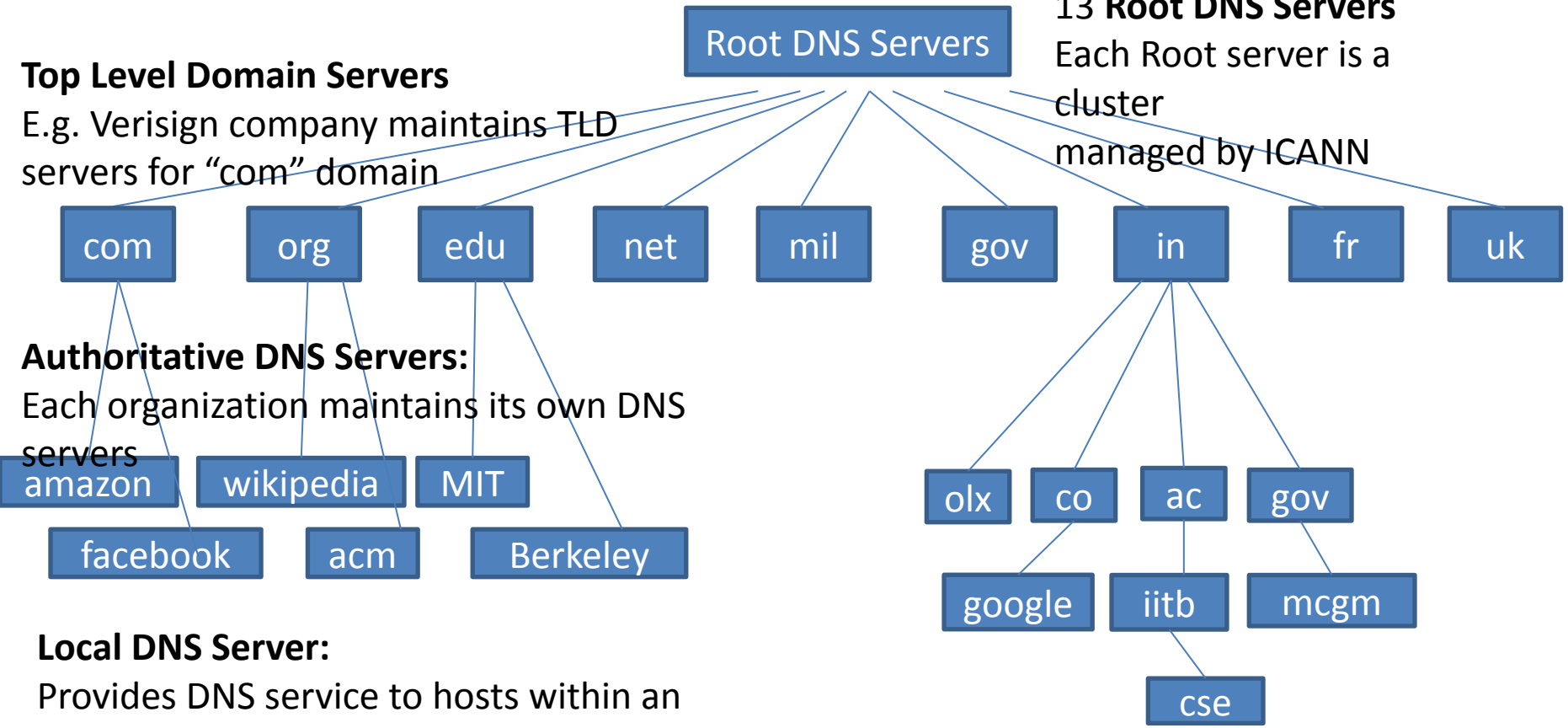
DNS: Problem and Solution

- People prefer hostnames
- Routers prefer IP addresses
- Need a service (DNS) that converts hostnames/domains to <u>Values</u>

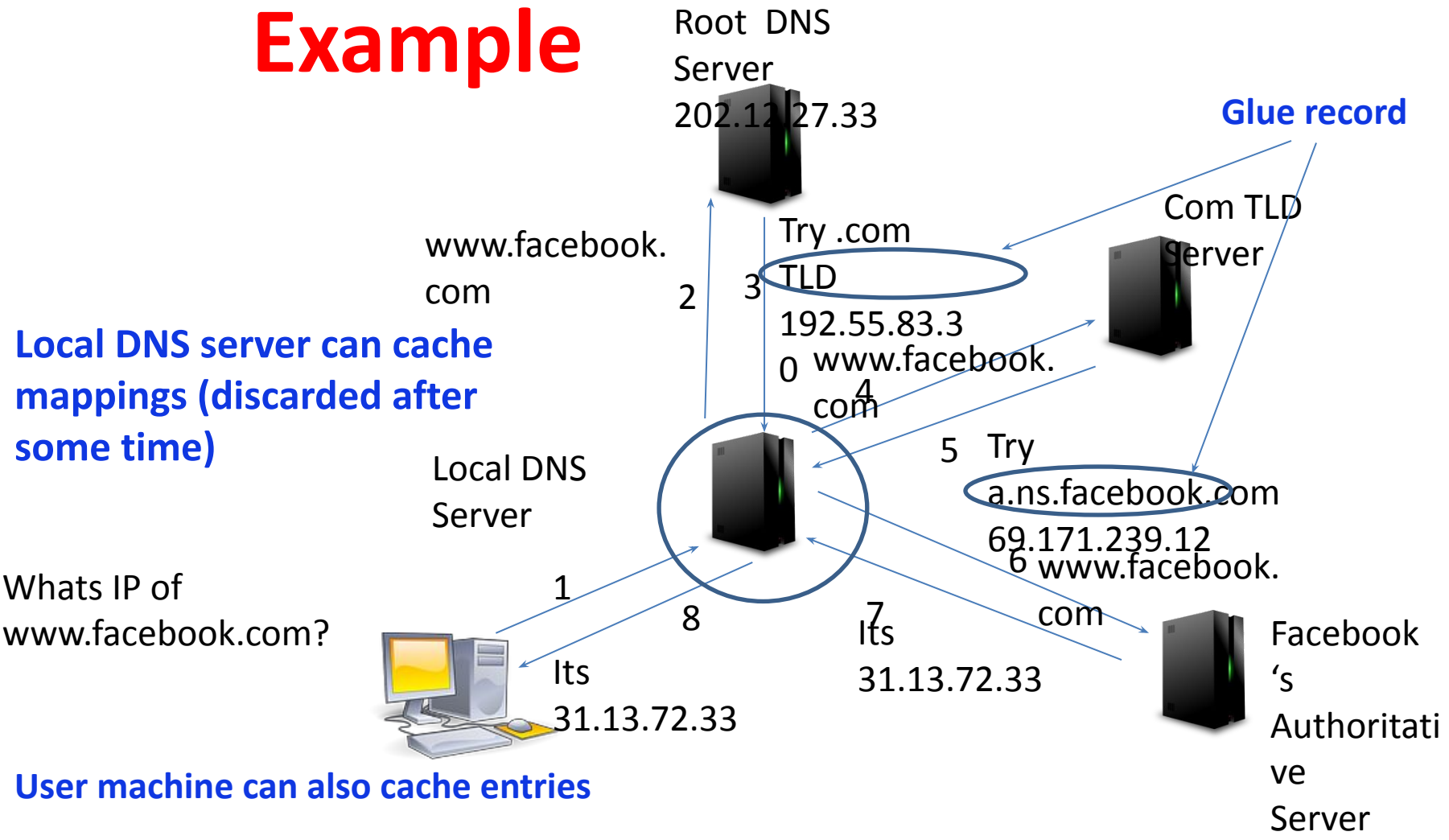


Domain Name: Label that defines a realm of administrative autonomy
E.g. facebook.com; iitb.ac.in; mit.edu

Hierarchical and Distributed Implementation



Example



Local DNS server can cache mappings (discarded after some time)

User machine can also cache entries

DNS Server Database

- Store Resource Records (RRs)
- Four Tuple: [Name, Value, Type, TTL]
- Type=A; Name: Hostname; Value: IP Address
  - E.g. [star.c10r.facebook.com, 31.13.72.33, A, 17]
- Type=NS; Name: Domain; Value: host-name of the authoritative name server
  - E.g. [facebook.com, a.ns.facebook.com, NS, 172797]

DNS Database

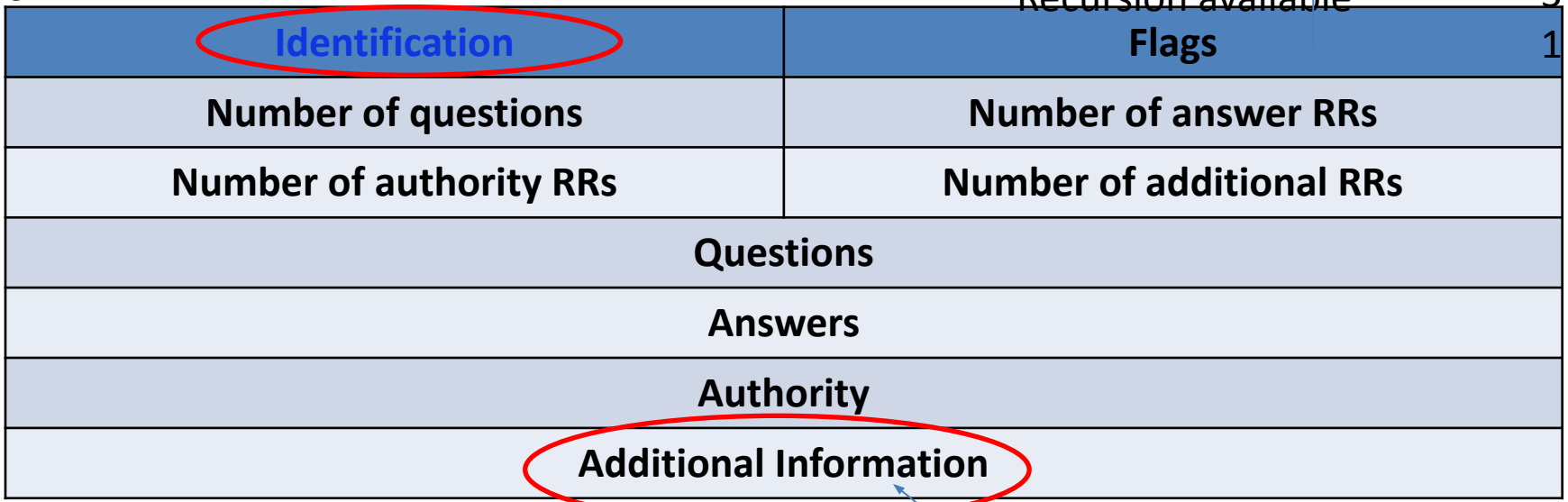
- Type=CNAME; Name: Hostname; Value: Canonical hostname
  - E.g. [www.facebook.com, star.c10r.facebook.com, CNAME, 2362]
- Type=MX; Name: Hostname; Value: Canonical name of the mail server
  - E.g. [facebook.com, msgin.t.facebook.com, MX, 300]

Rules

- An authoritative name server (for a given host) will always contain type A record of that host
- A non-authoritative name server will contain a type NS record for the domain and the type A record of the domain's authoritative server
  - [facebook.com, a.ns.facebook.com, NS, 172797]
  - [a.ns.facebook.com, 69.171.239.12, A, 172575]
- Demo: Dig command (see video)

DNS Message Format

Query/reply; Authoritative flag;
Recursion desired;
Recursion available



DNS runs over UDP and uses port 53

DNS Vulnerabilities

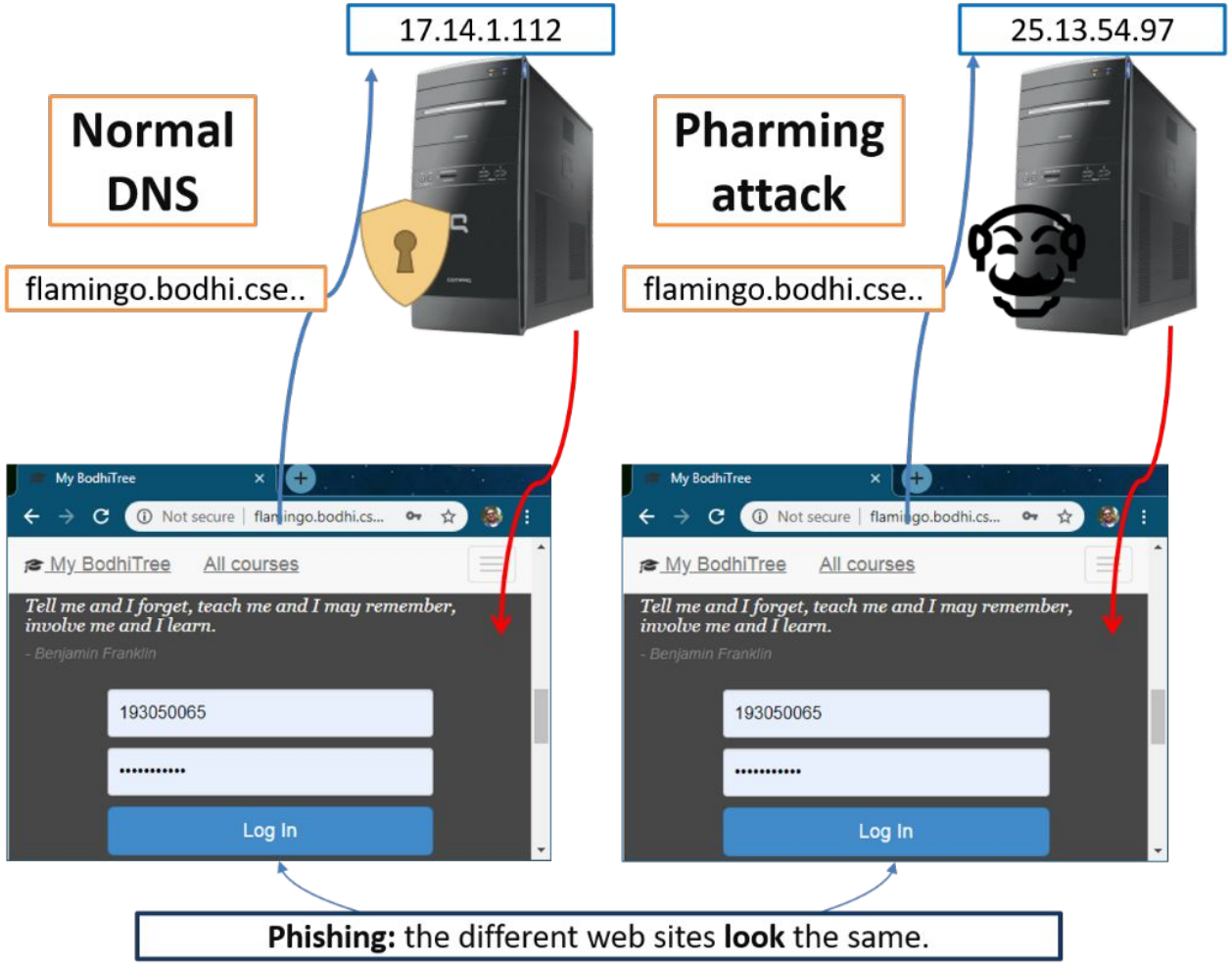
- No authentication of DNS responses
  - Relies solely on a 16-bit identification field
- Can insert fake records in cache via Glue records

Attacks: Pharming and Phishing

- Pharming: Names resolves to false values (of malicious host)
  - Very dangerous; DNS core service in Internet
  - When cached in local DNS, many downstream clients affected
  - Host can be web server, mail server, OS update server
- Web server: Phishing is where false website is near identical to original website
  - Malicious host can steal info, pass on malware
  - No easy way to detect

Attacks: Pharming and Phishing

- Mail server pharming can access mails
  - Passwords recovery of many sites often happens via emails
- OS update server pharming
  - Can pass on malicious code


17.14.1.112
25.13.54.97
Normal DNS
Pharming attack
flamingo.bodhi.cse..
flamingo.bodhi.cse..
My BodhiTree
Not secure
flamingo.bodhi.cs...
My BodhiTree
All courses
Tell me and I forget, teach me and I may remember, involve me and I learn.
- Benjamin Franklin
193050065
Log In
Phishing: the different web sites look the same.

How is Pharming done?

Many ways….

- **Rogue DNS server:**
  Suppose DNS server of iitd turned rogue. How can it poison cache and capture web traffic of say iitb ?

```
;QUESTION
www.iitd.ac.in. IN A
;ANSWER
www.iitd.ac.in. 8600 IN A 103.27.9.20
;AUTHORITY
iitd.ac.in. 8600 IN NS dns10.iitd.ac.in.
iitd.ac.in. 8600 IN NS dns8.iitd.ac.in.
;ADDITIONAL
dns8.iitd.ac.in. 8600 IN A 103.27.8.1
dns10.iitd.ac.in. 8600 IN A 103.27.10.1
```

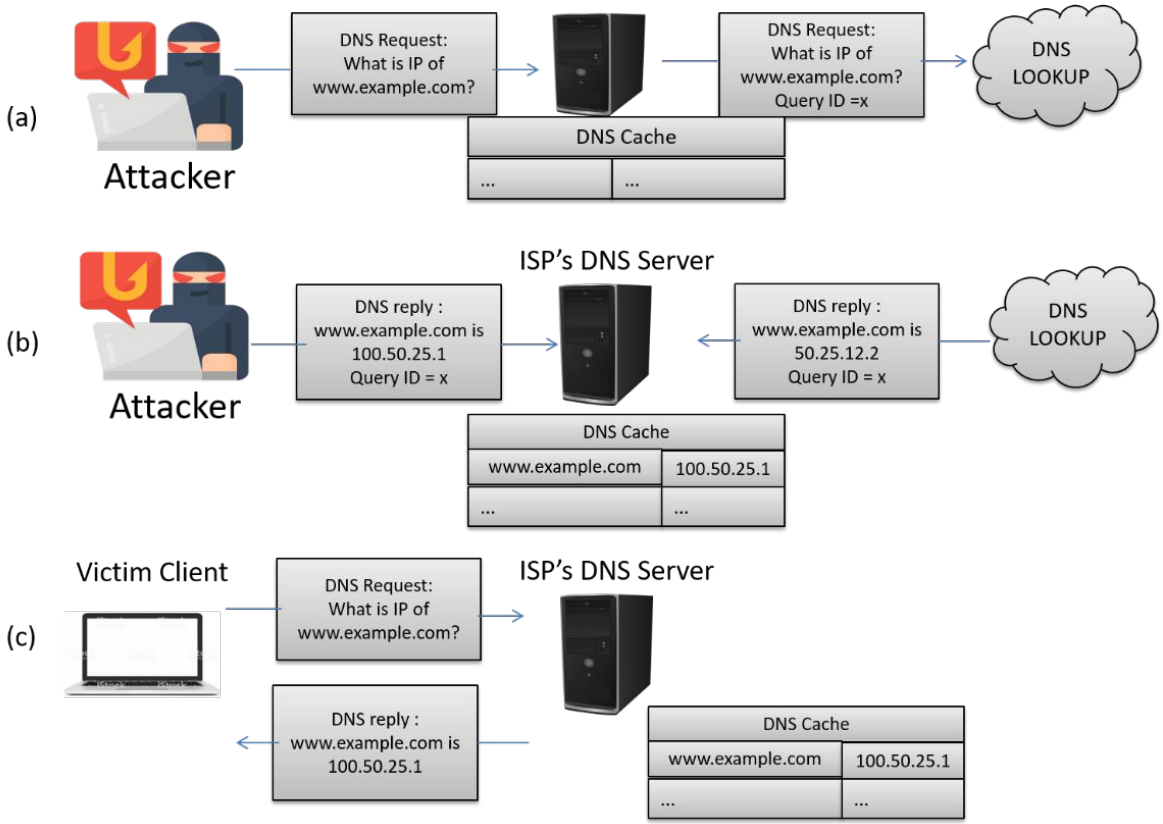
- Suppose a user (anywhere) contacts its local DNS to resolve www.iitd.ac.in
- Local DNS contacts DNS server of iitd (rogue)
- Reply from rogue DNS →
- 105.2.10.5 is a malicious web server (phising)
- Local DNS caches www.iitb.ac.in to 105.2.10.5 (attacker's web site) for 8600 sec (can be set longer also)
- All clients of 'local DNS' when they want to reach www.iitb.ac.in, land up on attacker's site

```
;QUESTION
www.iitd.ac.in. IN A
;ANSWER
www.iitd.ac.in. 8600 IN A 103.27.9.20
;AUTHORITY
iitd.ac.in. 8600 IN NS dns10.iitd.ac.in.
iitd.ac.in. 8600 IN NS www.iitb.ac.in.
;ADDITIONAL
www.iitb.ac.in. 8600 IN 105.2.10.5
dns10.iitd.ac.in. 8600 IN A 103.27.10.1
```

- Solution: Don’t accept additional records unless the belong to the same domain

On-Path DNS Attack

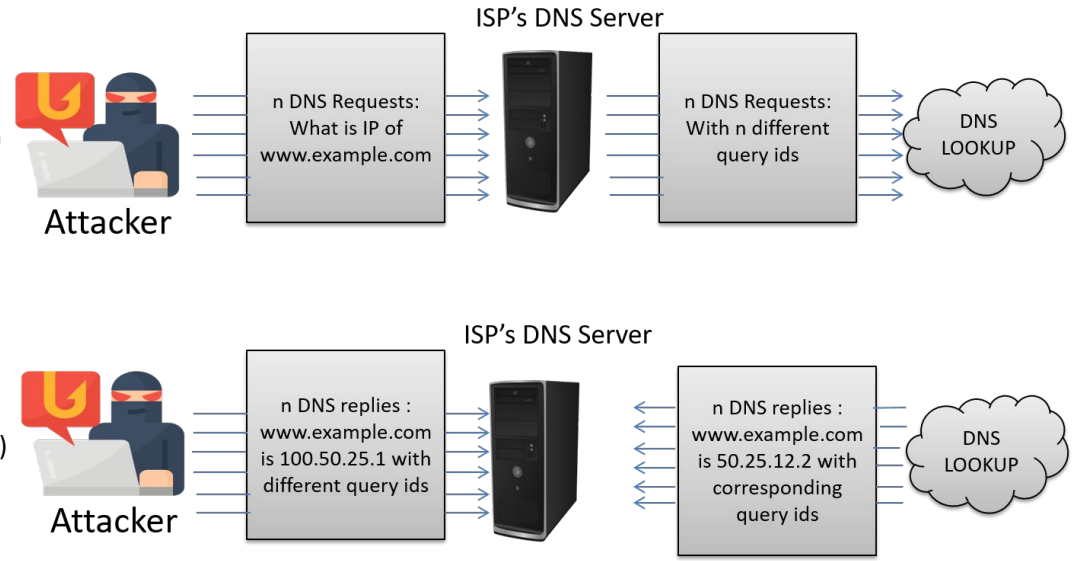
- Attacker wants to poison cache of an ISP's DNS server
- Attacker can sniff packets (DNS requests) sent by ISP's DNS server
- Attack Details: Can easily spoof a DNS reply
  - Sniffing requests (request id, Src/dest IP/port) helps construct appropriate reply
  - Attacker can trigger specific requests by querying the ISP's DNS server for the same
  - Attack succeeds only if spoofed DNS reply reaches ISP's DNS server faster than one from authoritative server


(a)
Attacker
DNS Request:
What is IP of
www.example.com?
DNS Request:
What is IP of
www.example.com?
Query ID =x
DNS
LOOKUP
DNS Cache
...
...
(b)
ISP's DNS Server
Attacker
DNS reply :
www.example.com is
100.50.25.1
Query ID = x
DNS reply :
www.example.com is
50.25.12.2
Query ID = x
DNS
LOOKUP
DNS Cache
www.example.com
100.50.25.1
...
...
(c)
Victim Client
DNS Request:
What is IP of
www.example.com?
ISP's DNS Server
DNS reply :
www.example.com is
100.50.25.1
DNS Cache
www.example.com
100.50.25.1
...
...

Off-Path (Blind) DNS Attack

- Guessing id tough (src/dst port often 53; IP addresses easy to figure out)
- Earlier DNS servers incremented id by 1 for every request
- Attack Details:
  - Send two DNS queries back to back (say www.evil.com and www.iitb.ac.in) to ISP's DNS server
  - First query will come to attacker's authoritative DNS for resolution , determine id x used
  - Spoof a reply to second query with id x+1
  - ISP's cache entry for www.iitb.ac.in poisoned (if spoofed reply faster)

- Solution: Use random id
- Birthday Paradox: Send large number of requests and fake replies
  - For N=213 (requests as well as fake replies), 50% chance one of the fake matches one of the requests
  - Challenge: race against time to beat replies from authoritative server
  - Authentic reply once cached, can be long wait before next attack



Sub-domain DNS Attack

- Discovered by Kaminsky
- Any way to avoid race against time?
- Issue many requests (N) for non-existent sub-domains (e.g. aaa.example.com, aab.example.com etc)
- Authoritative name server ignores such requests no race against time
- But only non-existent sub-domain poisoned. How does it help?

- Include a glue record
  - Name server of example.com maps to attacker's IP
  - Can alter name resolutions for the entire domain. Very dangerous!

Defences

- Most DNS attacks target local DNS servers local DNS servers should accept only internal requests
- Source port randomization: Apart from ID randomize the src port from which requests are made
  - Space: 2^16 possible ids * ~64000 possible ports
- This is what secures DNS today!
  - Not all resolvers have implemented source port randomization

DNSSEC

- Solutions are only stop gap measures, better approach secure DNS DNSSEC
- All DNS replies digitally signed
  - Based on chain of trust model
  - Root vouches .com; .com vouches for example.com; example.com vouches for another.example.com
  - Local DNS server (resolver) has the root's key wired in
  - All keys and certificates obtained are cacheable
- Requires changes to both client and server

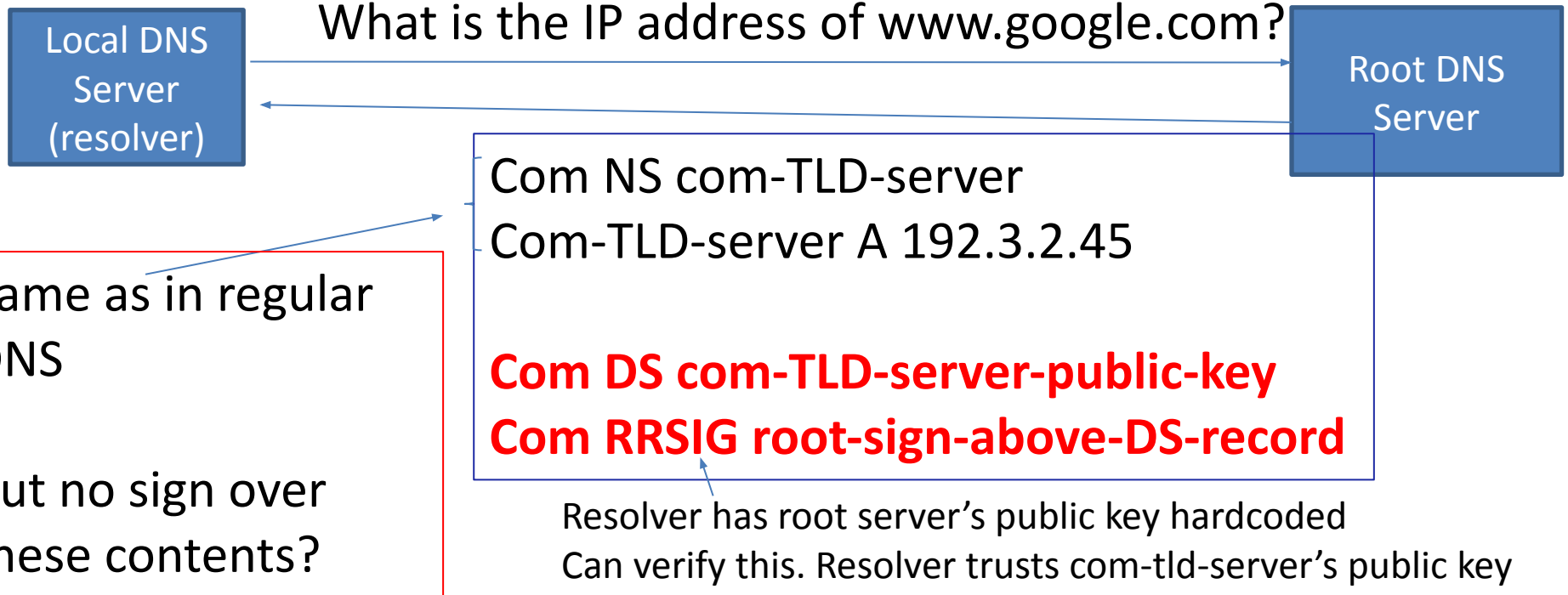
Records

Most relevant mentioned (there are more)

- **RRSIG (resource record signature)**
  - Contains the DNSSEC signature of hash of returned records
- **DS (delegation signer)**
  - Stored in the parent zone and contains info on child zone's public key

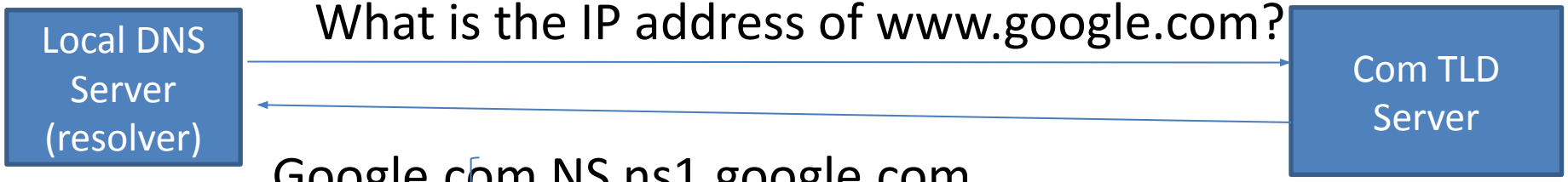
A Simplified View of DNSSEC

(Details will differ in actual implementation which is very complex)



A Simplified View of DNSSEC

(Details will differ in actual implementation which is very complex)

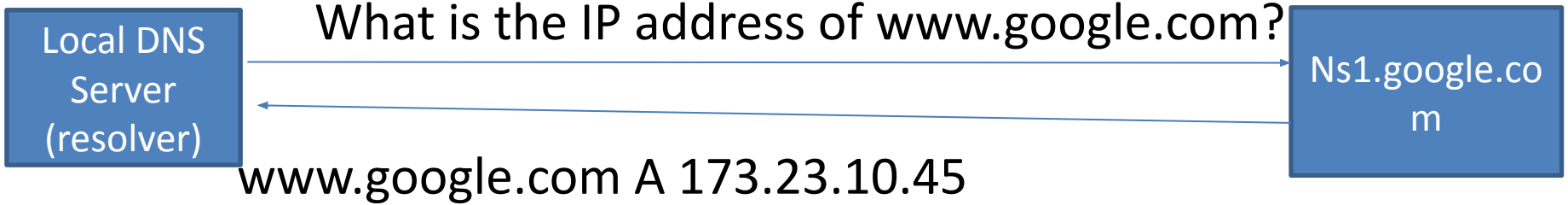


Google.com NS ns1.google.com
ns1.google.com A 173.4.7.45

Google.Com DS ns1-google-public-key
Google.Com RRSIG com-tld-server-sign-above-DS-record

A Simplified View of DNSSEC

(Details will differ in actual implementation which is very complex)



Deployment

- An ongoing deployment effort
- Partial deployment
  - Parent not supporting but child wants to? Not much can be done
  - Google.com supports but not facebook.com. Then should you accept DNS results for facebook?
    - Don’t accept: things break (facebook will be very upset)
    - Accept: weakens security; less incentive to migrate

Summary

- Looked at DNS based attacks
  - Randomization (port, ids) is more or less the current solution
- DNSSEC is the better option